



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

JÉSSICA SANTOS GUIMARÃES

PROTEÇÕES MECÂNICAS DOS MOTORES A DIESEL:
UM ESTUDO DE CASO E DESENVOLVIMENTO DE UM DISPOSITIVO DE
PROTEÇÃO CONTRA *OVERSPEED*

FORTALEZA

2013

JÉSSICA SANTOS GUIMARÃES

PROTEÇÕES MECÂNICAS DOS MOTORES A DIESEL:
UM ESTUDO DE CASO E DESENVOLVIMENTO DE UM DISPOSITIVO DE
PROTEÇÃO CONTRA *OVERSPEED*

Trabalho de conclusão de curso apresentada ao
Curso de graduação em Engenharia Elétrica da
Universidade Federal do Ceará, como requisito
parcial à obtenção do grau de Bacharel em
Engenharia Elétrica.

Orientador: Prof. Dr. Sérgio Daher.

FORTALEZA

2013

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca de Ciências e Tecnologia

G978p Guimarães, Jéssica Santos.
Proteções mecânicas dos motores a diesel: um estudo de caso e desenvolvimento de um dispositivo de proteção contra *overspeed* / Jéssica Santos Guimarães. – 2013.
119 f. : il. color., enc. ; 30 cm.

Monografia(graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia Elétrica, Graduação em Engenharia Elétrica, Fortaleza, 2013.
Orientação: Prof. Dr. Sérgio Daher.

1. Motor diesel. 2. Velocidade-controle automático. I. Título.

CDD 621.3

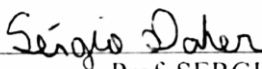
JÉSSICA SANTOS GUIMARÃES

PROTEÇÕES MECÂNICAS DOS MOTORES A DIESEL:
UM ESTUDO DE CASO E DESENVOLVIMENTO DE UM DISPOSITIVO DE
PROTEÇÃO CONTRA *OVERSPEED*

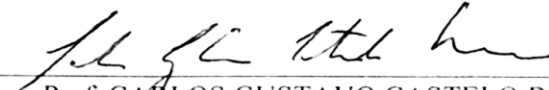
Trabalho de conclusão de curso apresentada ao
Curso de graduação em Engenharia Elétrica da
Universidade Federal do Ceará, como requisito
parcial à obtenção do grau de Bacharel em
Engenharia Elétrica.

Aprovada em: 12/12/2013.

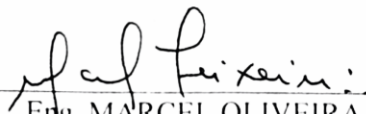
BANCA EXAMINADORA



Prof. SERGIO DAHER
(Orientador)



Prof. CARLOS GUSTAVO CASTELO BRANCO



Eng. MARCEL OLIVEIRA TEIXEIRA

A Deus.

Aos meus pais, professores, participantes da
minha formação técnica e amigos.

AGRADECIMENTOS

Aos meus pais que nunca sequer reclamaram o fato de ter na sala minha oficina e na área de serviço meu laboratório de confecção de placas de circuito impresso. Me apoiaram em todos os momentos.

À UFC, que me acolheu por todos estes anos nos quais andei ocupando seus laboratórios e salas de aula.

Ao Prof. Dr. Sergio Daher, pela excelente orientação.

Ao futuro colega de profissão e amigo Marcel Oliveira Teixeira que por inúmeras vezes me cedeu seu precioso tempo compartilhando conhecimentos.

Ao professor Carlos Gustavo Castelo Branco, participante da banca examinadora e responsável por boa parte do conhecimento o qual estarei a carregar pelos dias que se seguem.

A todos os professores que me receberam em suas salas, emprestaram livros, compartilharam conhecimento, forneceram materiais diversos e sempre demonstraram que desejam que eu tenha uma carreira profissional brilhante.

A AMP Serviços e Soluções Elétricas, empresa na qual eu pude ter a oportunidade de adquirir conhecimentos práticos muitos deles apresentados ao longo deste trabalho.

A todos os meus amigos pelas valiosas colaborações, sugestões e, acima de tudo, incentivo.

“A coragem para desafiar dificuldades é uma prática diária, assim como a opção pela evolução é uma escolha histórica.” (Capitão-de-Mar-e-Guerra Luiz Augusto Oliveira de Freitas)

RESUMO

Os principais acontecimentos que acarretam a redução da vida útil ou até falhas catastróficas em um motor a diesel são: perda de pressão do óleo lubrificante, elevação da temperatura da água do sistema de refrigeração e sobrevelocidade (*overspeed*). O monitoramento que previne a ocorrência de qualquer um destes é feito através de sensores que emitem sinais analógicos ou digitais para um dispositivo que controla a válvula de combustível, permitindo que o motor continue operando ou inicie uma parada de emergência. Este trabalho de conclusão de curso tem como objetivo a apresentação de um estudo de caso da instalação de um controlador que assegura todas as proteções mecânicas que um motor a diesel acoplado a um gerador deve possuir, segundo a norma IEEE *Standard* 45 (2002). Além disso, também é feito neste trabalho a exposição de todas as etapas necessárias e alcançadas para o desenvolvimento de um equipamento cuja função é a proteção do motor contra situações de sobrevelocidade (*overspeed*). O equipamento proposto inicia uma parada de emergência quando a velocidade excede 110% da velocidade nominal e o valor desta velocidade pode ser captado por um sensor magnético que detecta a cremalheira acoplada ao volante. Tanto o controlador escolhido e instalado como o equipamento desenvolvido foram ensaiados, através de simulações de situações de falhas, e apresentaram os resultados esperados. A escolha entre qual esquema de proteção escolher depende de muitos fatores, dentre os quais se pode citar o custo como sendo um fator decisivo.

Palavras-chave: Proteções mecânicas. Sobretemperatura da água. Pressão do óleo. Sobrevelocidade. Sensor Magnético.

ABSTRACT

The main events that result in the reduction of their working life or even catastrophic failures in a diesel engine are: loss of lube oil pressure, high water temperature of the cooling system and overspeed. The monitoring that prevents the occurrence of any of these is done through sensors that emit analog or digital signals to a device that controls the fuel valve, allowing the engine to continue to operate or start an emergency stop. This final paper aims at the presentation of a case study of installation of a controller that ensures all mechanical protections that a diesel engine coupled to a generator must possess, according to IEEE Standard 45 (2002). Furthermore, this work is also done exposing all the necessary and reached steps for the development of a device whose function is to protect the engine against overspeed situations. The proposed equipment initiates an emergency stop when the speed exceeds 110 % of rated speed and value of this speed can be picked up by a magnetic sensor that detects the rack attached to flywheel. Both the driver chosen and installed as the equipment developed were tested through simulations of situations of failures, and showed the expected results. The choice of which protection scheme choose depends on many factors, among which we can mention the cost as being a decisive factor.

Keywords: Mechanical protections. High water temperature. Oil Pressure. Overspeed. Magnetic Sensor.

LISTA DE ILUSTRAÇÕES

Figura 1 - Esquemático do Cilindro de um Motor Diesel.	23
Figura 2 - Esquemático do Ciclo Diesel de quatro tempos.	24
Figura 3 - Ligação mecânica entre cremalheira, pinhão e pistão.	25
Figura 4 - Sistema de lubrificação.	26
Figura 5 - Esquemático do Sistema de Resfriamento.	27
Figura 6 - Esquemático de um Sistema de Resfriamento para embarcações.	28
Figura 7 - Esquemático do Sistema de Partida.	29
Figura 8 - Esquemático de um gerador síncrono monofásico.	30
Figura 9 - Esquemático do retificador trifásico para bobinas do gerador ligadas em delta e estrela.	30
Figura 10 - Balanço de velocidade referenciando controle de velocidade.	32
Figura 11 - Sensor de Temperatura (termistor).	35
Figura 12 - Interruptores térmicos.	35
Figura 13 - Sensor de pressão de óleo	36
Figura 14 - <i>Pickup</i> magnético.	37
Figura 15 - Detalhes construtivos do <i>pickup</i> magnético e da cremalheira.	37
Figura 16 - Rebocador.	39
Figura 17 - Gerador modelo C110 D6 4 do fabricante Cummins.	39
Figura 18 - Controlador InteliDrive Lite da ComAp.	40
Figura 19 - Tela de controle e monitoramento do <i>software</i> LiteEdit da ComAP.	41
Figura 20 - Tela de configuração do <i>software</i> LiteEdit da ComAP.	42
Figura 21 - Botão <i>fault reset</i> do controlador InteliDrive Lite da ComAp.	45
Figura 22 - Botão <i>horn reset</i> do controlador InteliDrive Lite da ComAp.	45
Figura 23 - Tela de configuração das entradas digitais no <i>software</i> LiteEdit da ComAP.	46
Figura 24 - Tela de configuração das entradas analógicas no <i>software</i> LiteEdit da ComAP.	47
Figura 25 - Curva característica definida para o sensor de pressão no <i>software</i> LiteEdit da ComAP.	47
Figura 26 - Curva característica definida para o sensor de temperatura no <i>software</i> LiteEdit da ComAP.	48
Figura 27 - Esquemático do primeiro protótipo.	52
Figura 28 - Foto da placa utilizada para desenvolvimento do projeto.	53

Figura 29 - Programador PICKit 3.	55
Figura 30 - Gerador de frequência desenvolvido para testes.....	60
Figura 31 - Painel com sinaleiros de 12 VCC/VCA do fabricante ABB.	61
Figura 32 - Esquema de ligação utilizado nos ensaios.	63
Figura 33 - Bancada com <i>pickup</i> , motor e cremalheira.	64
Figura 34 - Sinal de onda coletado pelo osciloscópio com ponteira nos terminais do <i>pickup</i> sem que ele esteja conectado na placa.	65
Figura 35 - Sinal de onda coletado pelo osciloscópio com ponteira nos terminais do <i>pickup</i> com este inserido no circuito nos terminais RPM+ e RPM-.	65
Figura 36 - Sinal de onda coletado pelo osciloscópio com ponteira nos terminais de saída do optoacoplador (ente o pino T1CK e o GND).	66
Figura 37 - Leitura feita pelo protótipo paralelamente as coletas do osciloscópio.	67
Figura 38 - Visualização frontal do produto proposto.....	71
Figura 39 - Esquemático do projeto com o controlador InteliDrive Lite.	81
Figura 40 - Painel do grupo gerador com controlador da ComAp.	82
Figura 41 - Controlador InteliDrive Lite da ComAp operando em condições nominais.	82
Figura 42 - <i>Pickup</i> magnético instalado na carcaça do motor.	83
Figura 43 - Sensor de pressão instalado na carcaça do motor.	83
Figura 44 - Sensor de temperatura instalado na carcaça do motor.	84
Figura 45 - Lista de alarme apresentando, respectivamente, <i>shutdown</i> por perda de pressão do óleo e sobretemperatura da água.	84
Figura 46 Lista de alarme apresentando, respectivamente, <i>shutdown</i> por <i>overspeed</i> e parada de emergência.	84
Figura 47 - <i>Displays</i> de apresentação das variáveis monitoradas pelo controlador.	84
Figura 48 - Esquema de ligação utilizado nos ensaios.	85
Figura 49 - Simulação da condição nominal (frequência gerada equivalente ao motor girando em 1800 rpm).	86
Figura 50 - Simulação do motor operando com 500 rpm.....	87
Figura 51 - Simulação de uma situação de <i>overspeed</i> de exatamente 110%.	88
Figura 52 - Esquemático da placa final produzida no <i>Altium Designer</i> com o dsPIC33FJ12MC202.	89
Figura 53 - <i>Layout</i> da placa de circuito impresso contendo o dsPIC33FJ12MC202 produzido no Altium Designer.	90

Figura 54 - Vista superior gerada através do <i>SolidWorks</i> e desenvolvida no <i>Altium Designer</i> com o dsPIC33FJ12MC202.	90
Figura 55 - Vista inferior gerada através do <i>SolidWorks</i> e desenvolvida no <i>Altium Designer</i> com o dsPIC33FJ12MC202.	91
Figura 56 - Regulador de tensão com saída de 3,3 V.	91
Figura 57 - Esquemático produzido no software <i>Proteus</i> para simular os códigos para o dsPIC33FJ12MC202.	94
Figura 58 - Resultados de simulação apresentados pelo software <i>Proteus</i>	95
Figura 59 - Resultados de simulação apresentados pelo software <i>Proteus</i> em uma situação de falha.	95
Figura 60 - Vista frontal do produto final.	96
Figura 61 - Vista posterior do produto final.	96
Figura 62 - Visualização do frontal do dispositivo.	97
Figura 63 - Visualização da parte posterior do dispositivo.	97
Figura 64 - Detalhe dos bornes de conexão e dos potenciômetros existentes no dispositivo. .	98
Figura 65 - Folheto desenvolvido para o dispositivo proposto.	98
Figura 66 - Circuito de comando com as proteções necessárias aos motores.	99

LISTA DE TABELAS

Tabela 1 - Resumo das saídas digitais utilizadas no controlador InteliDrive Lite.	79
Tabela 2 - Resumo das entradas digitais utilizadas no controlador InteliDrive Lite.....	79
Tabela 3 - Resumo das entradas analógicas utilizadas no controlador InteliDrive Lite.....	79
Tabela 4 - Resumo dos valores das principais variáveis do controlador InteliDrive Lite.....	80
Tabela 5 - Lista de materiais necessários.	92

LISTA DE CÓDIGOS

Código 1 - Arquivo "main.h" para o dsPIC 30F4011.....	100
Código 2 - Arquivo "timer_x.h" para o dsPIC 30F4011.	101
Código 3 - Arquivo "config_interrupts.h" para o dsPIC 30F4011.	102
Código 4 - Arquivo "interrupts.h" para o dsPIC 30F4011.	103
Código 5 - Arquivo "LCD.h" para o dsPIC 30F4011.....	105
Código 6 - Arquivo "main.c" para o dsPIC 30F4011.....	108
Código 7 - Arquivo "main.h" para o dsPIC33FJ12MC202.....	110
Código 8 - Arquivo "config_interrupts.h" para o dsPIC33FJ12MC202.	111
Código 9 - Arquivo "interrupts.h" para o dsPIC33FJ12MC202.	112
Código 10 - Arquivo "timer_x.h" para o dsPIC33FJ12MC202.	113
Código 11 - Arquivo "LCD.h" para o dsPIC33FJ12MC202.....	114
Código 12 - Arquivo "main.c" para o dsPIC33FJ12MC202.	118

LISTA DE ABREVIATURAS E SIGLAS

FMM	Fundo de Marinha Mercante
GEICON	Grupo Executivo da Indústria de Construção Naval
CMM	Comissão da Marinha Mercante
PROMEF	Programa de Modernização e Expansão da Frota
ABENAV	Associação Brasileira das Empresas de Construção Naval e <i>Offshore</i>
ABS	<i>American Bureau of Shipping</i>
BC	<i>Buareau Colombo</i>
BV	<i>Bureau Veritas</i>
RINA	<i>Registro Italiano Navale</i>
IACS	<i>International Associations of Classifications Societies Ltd</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
MCA	Motor de Combustão Auxiliar
MCP	Motor de Combustão Principal
INACE	Indústria Naval do Ceará
IMO	International Maritime Organization
MPU	<i>Magnetic Pickup Unit</i>
IHM	Interface Homem-Máquina
MIPS	Milhões de Instruções Por Segundo
CM	Ciclo de Máquina

LISTA DE SÍMBOLOS

$f_{alternador}$	Frequência da tensão gerada pelo alternador [Hz]
n_{motor}	Velocidade do motor [rpm]
$p_{alternador}$	Número de pólos do alternador
f_{pickup}	Frequência da tensão gerada pelo sensor de velocidade [Hz]
n_{teeth}	Número de dentes da cremalheira
F_{cy}	Frequência de execução das instruções de código [Hz]
F_{osc}	Frequência do <i>clock</i> fornecido ao dsPIC [Hz]
PLL	Multiplicador da frequência do <i>clock</i> fornecido ao dsPIC
CM	Ciclo de máquina [s]
T_{cy}	Período do <i>clock</i> da máquina [s]
$PR1$	Valor do estouro do <i>timer</i> 1
$PR2$	Valor do estouro do <i>timer</i> 2
T_{timer1}	Período do <i>timer</i> 1 [s]
T_{timer2}	Período do <i>timer</i> 2 [s]
$prescale$	Divisor dos contadores do <i>timers</i>
$CONTAGEM$	Número de bordas de subida apresentadas em 500 ms
M	Múltiplo calculado a partir do número de dentes da cremalheira
V_{limite}	Velocidade a partir da qual a proteção deve atuar [rpm]
V_n	Velocidade nominal de operação do motor [rpm]
$AJUSTEFIXO$	Valor que delimita as condições de proteção a 110%
V_{medida}	Valor da velocidade do motor calculada pelo dsPIC [rpm]

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Estrutura do trabalho	21
2	MOTORES DE COMBUSTÃO INTERNA	22
2.1	Princípio de funcionamento dos motores a diesel.....	23
2.2	Componentes do motor de combustão interna a diesel.....	24
2.3	Sistemas dos motores de combustão a diesel.....	25
3	NORMATIZAÇÕES	33
4	SENSORES	34
4.1	Sensor de temperatura	34
4.2	Sensor de pressão.....	35
4.3	Sensor de velocidade.....	36
5	ESTUDO DE CASO	39
5.1	Controlador	40
5.2	Configurações	41
5.2.1	<i>Saídas digitais</i>	<i>43</i>
5.2.2	<i>Entradas digitais</i>	<i>45</i>
5.2.3	<i>Entradas analógicas</i>	<i>46</i>
5.2.4	<i>Sensor de velocidade.....</i>	<i>48</i>
5.3	O projeto utilizando o controlador InteliDrive Lite da ComAp	49
5.4	Ensaios	49
6	PROJETO DE UM DISPOSITIVO DE PROTEÇÃO CONTRA OVERSPEED	51
6.1	O primeiro protótipo	51
6.1.1	<i>Resumo dos componentes.....</i>	<i>53</i>
6.1.2	<i>Códigos.....</i>	<i>54</i>

6.1.3 Ensaio com gerador de frequência.....	60
6.1.4 Ensaio com pickup magnético	63
6.2 O segundo protótipo	68
6.2.1 A placa de circuito impresso	68
6.2.2 Códigos.....	69
6.2.3 Simulações	70
6.2.4 Produto final	71
6.2.5 Exemplo de aplicação	72
7 CONCLUSÃO.....	74
7.1 Sugestões para trabalhos futuros	74
REFERÊNCIAS	76
APÊNDICE A - Resumo das configurações e esquemático das ligações elétricas definidas para o controlador InteliDrive Lite.	79
APÊNDICE B - Imagens do painel no qual foi instalado o controlador InteliDrive Lite.	82
APÊNDICE C - Ensaio realizado com gerador de frequência inserindo o sinal na placa com o dsPIC30F4011.	85
APÊNDICE D - Arquivos utilizados para o desenvolvimento do dispositivo utilizando o dsPIC33FJ12MC202.	89
APÊNDICE E - Exemplo de aplicação.	99
APÊNDICE F - Códigos utilizados no dsPIC 30F4011.....	100
APÊNDICE G - Códigos utilizados no dsPIC33FJ12MC202.....	110

1 INTRODUÇÃO

A indústria naval brasileira surgiu no final do século XIX quando foi construído o primeiro estaleiro, em 1847, no Rio de Janeiro, e teve um primeiro auge na década de 1970 quando o Brasil chegou a ser o segundo maior parque naval do mundo devido a incentivos governamentais. Dentre estes incentivos, podem-se citar os Planos de Construção Naval e a criação de leis como a do Fundo de Marinha Mercante (FMM) em 1958 que tinha como objetivos criar fundos destinados a prover recursos para renovação, ampliação e recuperação da frota mercante nacional, estimular a exportação de embarcações, entre outros. Nesta mesma época, também foram criados o Grupo Executivo da Indústria de Construção Naval (GEICON) e a Comissão da Marinha Mercante (CMM) (JESUS; GITAHY, 2009).

Logo após este período, nas décadas de 1980 e 1990, devido a problemas políticos e econômicos, a indústria naval sofreu uma crise e a diminuição da produção foi inevitável devido à concorrência com os estaleiros asiáticos que conseguiam manter preços mais atraentes aos armadores. Armador nada mais é do que uma entidade que faz a exploração comercial de um navio, sendo ou não proprietário deste.

A indústria naval só veio apresentar novamente crescimento no final da década de 1990 com a necessidade da exploração *offshore* de petróleo. Devido a isto foi criada a Lei do Petróleo, o Programa Navega Brasil e o Programa de Modernização e Expansão da Frota (PROMEF) da TRANSPETRO.

Diante deste mercado de exploração de petróleo, surgiu uma demanda por novas plataformas, novos estaleiros, navios e embarcações de apoio marítimo. O PROMEF, programa do governo federal, ainda em execução, por exemplo, tem como premissa a construção de 49 navios e estaleiros que se tornem mundialmente competitivos. Nos últimos três anos, três estaleiros já foram criados (Atlântico Sul, Vard Promar e Rio Tietê), ou seja, o cenário da indústria naval está novamente fomentado e chega a empregar atualmente 70 mil pessoas, conforme informações da empresa TRANSPETRO (2013).

A indústria naval e os fabricantes de equipamentos para o setor nada mais fazem do que acompanhar o interesse do mercado. A Associação Brasileira das Empresas de Construção Naval e *Offshore* (ABENAV) (2013) prevê a compra pela Petrobrás de mais de 100 unidades de plataformas de produção e sondas de perfuração e mais de 500 barcos de apoio, mais de 80 petroleiros e muitos outros equipamentos.

Paralelamente a esta fabricação de embarcações, um processo importante que deve ser citado é o registro do navio o qual está associado ao Estatuto Legal, que é obrigatório e é atribuído pela Administração do País de bandeira do navio e ao Estatuto de Classe que é conferido pelas denominadas Sociedades Classificadoras, entidades que atuam em nome da Autoridade Marítima Brasileira fazendo o controle, a regularização e a certificação das embarcações para que sejam garantidas a segurança da navegação, a prevenção da poluição ambiental e a salvaguarda da vida humana.

O fato de uma embarcação estar classificada dita algumas vantagens dentre as quais se podem citar: o valor do seguro a ser calculado pela seguradora que tem valor menor; maior atratividade no mercado internacional; valorização do navio e reconhecimento pelas Autoridades Marítimas (EMÍLIO, 2009).

Algumas sociedades classificadoras podem ser citadas: *American Bureau of Shipping* (ABS), *Buareau Colombo* (BC), *Bureau Veritas* (BV), *Registro Italiano Navale* (RINA), *International Associations of Classiffications Societies Ltd* (IACS), dentre outras.

Para que seja emitido o certificado pela Sociedade Classificadora, regras definidas pelas normas de cada classificadora devem ser seguidas e só após a conclusão satisfatória das intensas inspeções o certificado pode ser emitido e o navio pode ser dito classificado. Para embarcações em operação, inspeções também são realizadas para se constatar que o navio permanece em conformidade com as regras estabelecidas pela classificadora, caso contrário se perde a classe.

As normas técnicas desenvolvidas pela classificadora visam a segurança do navio e são todas baseadas na *IEEE Standard 45* (2002), norma internacional que aborda requisitos técnicos da indústria marítima desde a seleção de equipamentos elétricos e eletrônicos até sua instalação.

Alguns sistemas principais devem ser apresentados de imediato para que se possa ter uma noção geral do estudo a ser feito. Uma embarcação possui o sistema de propulsão responsável pelo deslocamento da mesma e pode ser à vela, máquinas de combustão externa (turbina a vapor, por exemplo), máquinas de combustão interna (turbina a gás e motor diesel são exemplos), máquinas elétricas e ainda propulsão nuclear que é utilizada em submarinos.

Um outro sistema é o de produção de energia elétrica que é composto por geradores elétricos que são acionados ou por motores de combustão auxiliar (MCAs) ou pelo próprio motor de combustão principal (MCP) que é responsável pela propulsão. Cada um desses geradores é capaz de suportar a carga elétrica do navio em regime normal.

O enfoque deste trabalho está no estudo dos motores de combustão interna que são largamente utilizados tanto para produção de energia como para propulsão.

Com relação a estes equipamentos (MCAs, MCPs), tanto a norma IEEE 45 quanto as classificadoras determinam uma série de regras a serem cumpridas das quais se podem citar inicialmente as proteções necessárias para que possam ser evitadas avarias ao equipamento ou acidentes que possam vir a comprometer a segurança da tripulação ou da estrutura da embarcação.

São exigidas proteções contra *overspeed*, sobre temperatura da água de arrefecimento e baixa pressão do óleo lubrificante. Algumas proteções são ditas opcionais: baixo nível de óleo, baixo nível de água, *underspeed*, alta temperatura do óleo lubrificante, etc. A elevação excessiva de temperatura pode causar até a fusão de alguns metais e *overspeed*, que pode ser causada pela retirada de carga abruptamente, pela interrupção da excitação da máquina ou simplesmente uma falha no sistema de injeção de combustível, pode acarretar danos mecânicos à máquina.

Já foi explanado anteriormente sobre a demanda atual gerada para a indústria naval. Considerando o panorama local da cidade de Fortaleza, isto pode ser evidenciado pela Indústria Naval do Ceará (INACE), a qual possui grande produção de rebocadores e barcas. Os rebocadores são navios de grande robustez, alta potência e mobilidade que são utilizados para reboque ou prestar socorros tais como combate a incêndio e serviços de esgoto, já as barcas se destinam ao transporte marítimo de uma margem a outra de um rio (FONSECA, 2002, p. 126). Paralelamente a isto, uma demanda por serviços de manutenção e modernização destas embarcações que atuam no terminal portuário do Pecém e no Porto do Mucuripe (na cidade de Fortaleza) é gerada para empresas de instalações elétricas.

Um destes serviços diz respeito a instalação de controladores que ofereçam todas as proteções necessárias aos motores de combustão interna. O controlador InteliDrive da empresa ComAP é um exemplo de equipamento com tal capacidade logo, é com ele que será apresentado, ao longo deste texto, um estudo de caso desenvolvido nos dois MCAs de um rebocador.

Apesar da existência destes controladores, propõe-se neste trabalho o desenvolvimento de um dispositivo que viabiliza o monitoramento de velocidade e sinaliza a ocorrência de *overspeed* quando a velocidade medida é superior em 10% a velocidade nominal do motor. Para tal, com o auxílio de potenciômetros, deve ser feito o ajuste do valor das variáveis que definem as características de operação do motor (velocidade nominal e

número de dentes da cremalheira). Além disto, contatos normalmente abertos e fechados são disponibilizados para que possam ser inseridos em um circuito de comando que contemple todas as proteções necessárias.

A grande vantagem de tal desenvolvimento é poder fazer com que seja possível utilizar este dispositivo assim como se usa um sensor de temperatura da água que abre ou fecha um contato quando a temperatura de 92 °C é atingida ou um sensor de pressão que altera a posição do contato quando a pressão do óleo passa a ser menor que 1bar, por exemplo. Estes dispositivos em conjunto passariam a determinar o funcionamento do solenóide de combustível que comanda a parada e a partida do motor. Ou seja, um circuito de comando de baixo custo e alta eficiência pode ser desenvolvido com apenas alguns relés, sensores de temperatura e pressão, que são facilmente obtidos no mercado local e o dispositivo de proteção contra *overspeed*.

1.1 Estrutura do trabalho

Este trabalho é composto de sete capítulos. No primeiro capítulo foi dada uma introdução geral do assunto que será apresentado assim como pôde ficar clara a motivação do trabalho proposto.

No segundo capítulo é feita uma abordagem sobre o motor a diesel, seu princípio de funcionamento, os principais componentes que o constitui e seus sistemas (lubrificação, partida, resfriamento e de combustível, dentre outros).

No terceiro capítulo é dado enfoque as normatizações as quais devem ser seguidas logo, a título de conhecimento, são apresentados trechos da norma internacional IEEE 45 (2011) e das normas de sociedades classificadoras.

O capítulo 4 apresenta os sensores através dos quais é possível desenvolver as funções de monitoramento, controle e proteção do motor visto que eles transformam grandezas como temperatura, pressão e velocidade em medidas elétricas (resistência e tensão).

É no quinto capítulo onde pode ser visto o estudo de caso no qual foi escolhido um controlador da empresa ComAp, com o qual foi possível garantir a proteção do motor.

O sexto capítulo expõe toda a metodologia utilizada para desenvolver o dispositivo de proteção contra *overspeed*, detalha o protótipo e a proposta final, revela os resultados de ensaios e simulações e apresenta um exemplo de aplicação com o equipamento.

No último capítulo, conclusões e sugestões são externadas pela autora.

2 MOTORES DE COMBUSTÃO INTERNA

A combustão é o nome dado para a reação química que ocorre entre o oxigênio, denominado comburente, e o combustível que pode ser madeira, papel, tinta, algodão, álcool, gasolina, diesel, etc. Esta reação libera muita energia e é justamente esta energia térmica que é transformada em energia mecânica. Define-se como máquina de combustão o dispositivo que transforma a energia química de um combustível em energia térmica e usa esta energia para produzir trabalho mecânico (GUPTA, 2013).

Os motores de combustão podem ser classificados como sendo de combustão interna e externa. A diferença está justamente no local onde ocorre a reação química. Na máquina de combustão externa, o processo se dá fora dela, numa caldeira, por exemplo, e o calor gerado produz vapor d'água, que ao expandir, provoca o movimento da máquina. O fluido de trabalho está completamente separado da mistura de combustível e ar (TAYLOR, 1985). Como exemplo, pode-se citar a turbina a vapor que foi largamente utilizada por muitos anos como sistema de propulsão de navios.

Na máquina de combustão interna, o processo químico ocorre no interior da própria máquina. Segundo Gupta (2013), os motores de combustão interna podem ser divididos quanto ao seu ciclo termodinâmico (Otto, Diesel e Brayton), ao ciclo de trabalho (dois tempos e quatro tempos), ao tipo de combustível utilizado (motores de óleo leve utilizando querosene ou gasolina, motores de óleo pesado utilizando diesel ou óleo mineral, motores a gás ou motores que utilizam tanto o gás como combustível líquido), ao método de fornecimento de combustível, ao método de ignição, ao método de resfriamento (água ou ar), à velocidade desenvolvida, ao campo de aplicação, dentre outras variáveis.

Quanto ao campo de aplicação, os motores podem ser: estacionários quando utilizados para geração de energia, ou seja, a velocidade constante; motores marítimos de propulsão de navios; motores automotivos para transporte por terra; motores aeronáuticos e motores de locomotivas para vias férreas. Neste trabalho são objeto de estudo os motores estacionários que são acoplados a geradores que suportam toda a carga elétrica do navio e os motores de propulsão responsáveis pelo deslocamento da embarcação.

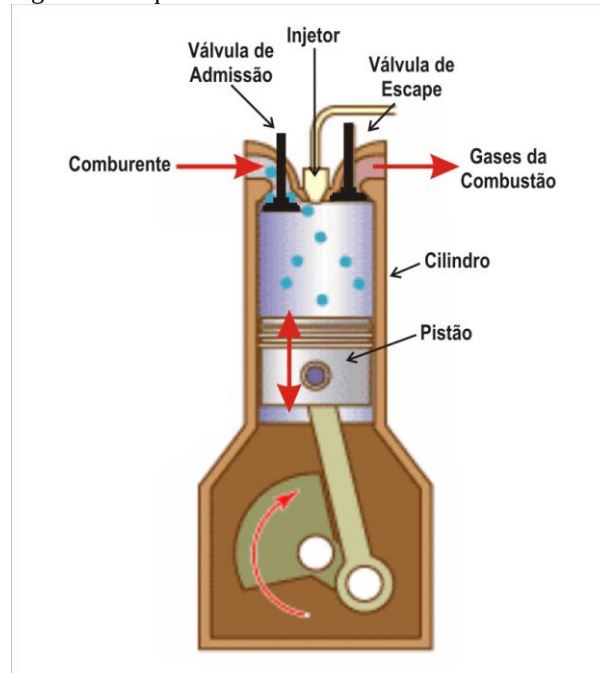
Embora tenham surgido problemas como o da redução das reservas de petróleo, e o surgimento de restrições aplicadas pela IMO (*International Maritime Organization*) com relação a limites para emissões de gases do efeito estufa, a maioria dos navios da frota

mundial ainda usa motores diesel tanto em suas instalações propulsoras como nos seus sistemas auxiliares (FILHO, 2011).

2.1 Princípio de funcionamento dos motores a diesel

Para entender como funciona o ciclo Diesel inicialmente devem ser citados alguns componentes básicos do motor os quais podem ser vistos na Figura 1. O cilindro é a peça no interior da qual ocorre todo o ciclo termodinâmico (transformações que se repetem com uma lei periódica e com variações de pressão, volume e temperatura). O pistão é o item que se move longitudinalmente no interior do cilindro sendo responsável, portanto, pelo movimento do motor. Através da válvula de admissão é possível a entrada ou não de comburente; a válvula de escape permite a saída de gases da combustão; e o injetor a entrada do combustível.

Figura 1 - Esquemático do Cilindro de um Motor Diesel.

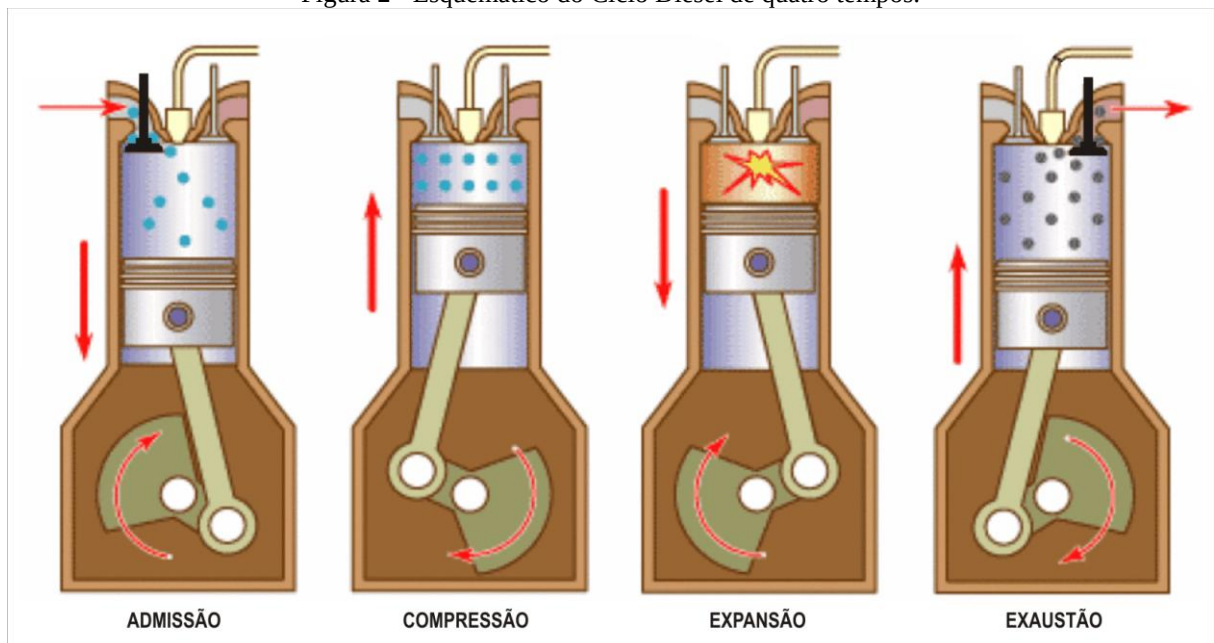


Fonte: <<http://tatofisicatermodinamica.wordpress.com>>
(Editada pelo próprio autor)

O ciclo termodinâmico Diesel pode ser de dois ou quatro tempos. O ciclo de 4 tempos possui as seguintes etapas: admissão, compressão, expansão e exaustão, conforme pode ser visto na Figura 2.

De acordo com Ganesan (2007), no período de admissão a válvula de admissão é aberta permitindo que uma certa quantidade de ar penetre no cilindro e desloque o pistão para o ponto inferior enquanto a válvula de escape permanece fechada. A segunda etapa é denominada compressão na qual ambas as válvulas estão fechadas e, devido ao movimento de retorno do pistão para a posição superior, é feita a compressão do ar admitido. Ao final da compressão, a temperatura do ar é suficiente alta para que ocorra ignição do combustível que passa a ser liberado pelo injetor. Ocorre a combustão e, como ambas as válvulas permanecem fechadas, os produtos da combustão conseguem expandir forçando o movimento do pistão novamente para baixo. Esta etapa é intitulada expansão e, ao final desta, a válvula de escape se abre permitindo que, devido a pressão, uma parte dos gases evacuem. Ao haver o deslocamento do pistão novamente para a posição superior, os produtos restantes da combustão são liberados para fora do cilindro. Todo o processo descrito pode ser melhor visualizado através da Figura 2.

Figura 2 - Esquemático do Ciclo Diesel de quatro tempos.



Fonte: < <http://tatofisicatermodinamica.files.wordpress.com/2013/05/12.gif> >
(Editada pelo próprio autor)

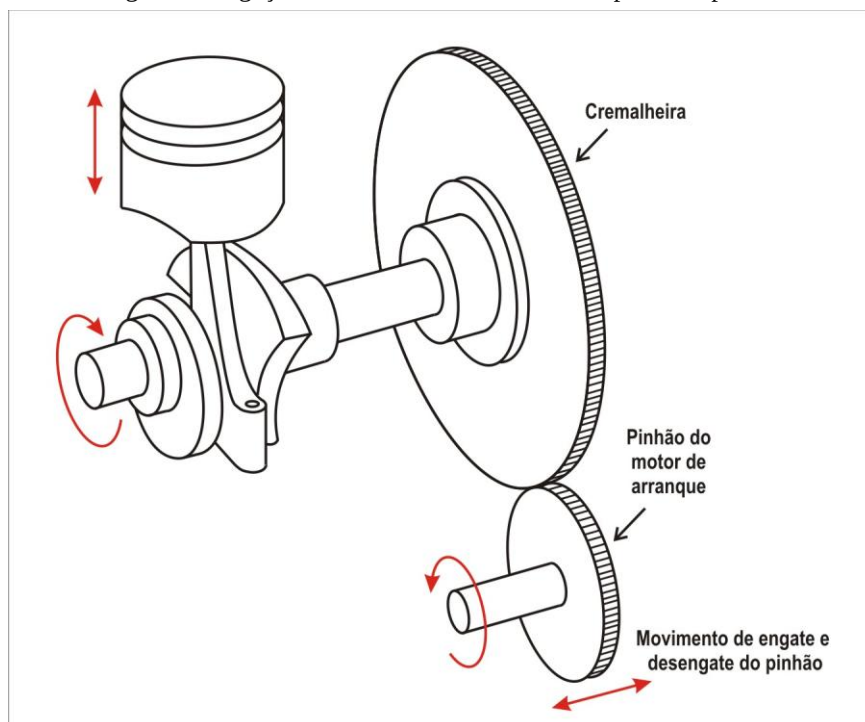
2.2 Componentes do motor de combustão interna a diesel

O motor a diesel é composto de peças fixas e móveis. O bloco é a maior peça fixa do motor e é nele onde serão alojados os cilindros; o cabeçote é fixado ao bloco funcionando como uma peça que fecha os cilindros e no qual são instaladas as válvulas de admissão e

escape e o injetor; o cárter é um depósito que tem como propósito o armazenamento de óleo lubrificante do motor (PEREIRA, 2006).

Um item de fundamental importância para este trabalho é o volante, um disco grande e pesado fixado ao eixo o qual tem por finalidade armazenar energia mecânica durante a expansão para vencer a resistência na compressão (MARINHA DO BRASIL, 2008, p. 22). O volante pode possuir uma coroa dentada denominada cremalheira através da qual é possível acoplar o pinhão do motor de arranque, utilizado somente na partida (Figura 3).

Figura 3 - Ligação mecânica entre cremalheira, pinhão e pistão.



Fonte: Produção do próprio autor.

2.3 Sistemas dos motores de combustão a diesel

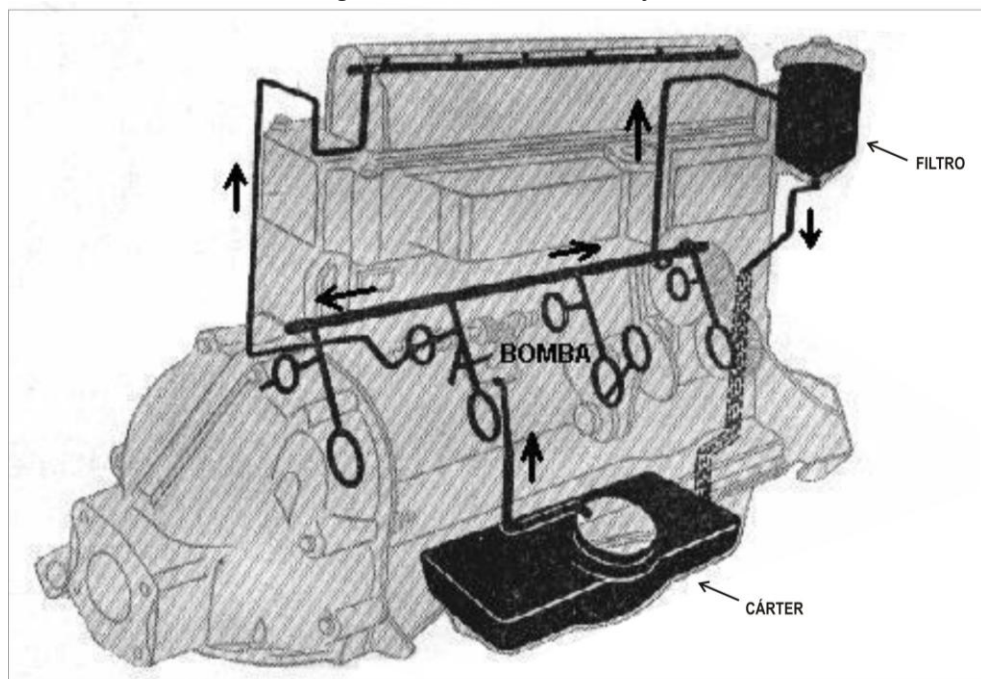
O motor de combustão interna não se resume simplesmente ao processo químico que ocorre dentro do cilindro, alguns sistemas são necessários para o funcionamento da máquina. São os sistemas: de lubrificação, de resfriamento, de combustível, de alimentação de ar, de partida, de geração de energia elétrica e o de descarga dos gases da combustão, entre outros.

- SISTEMA DE LUBRIFICAÇÃO

Toda máquina é composta por um conjunto mecânico o qual possui inúmeras peças móveis em contato gerando atrito, calor e, conseqüentemente, desgaste. O objetivo deste sistema de lubrificação é reduzir o atrito entre estas peças além de resfriá-las e limpá-las. Tanto o bloco como o cabeçote são vazados internamente para que haja um caminho de circulação do lubrificante que, como já foi dito, fica armazenado no cárter (Figura 4).

A bomba de óleo, que pode estar acoplada diretamente ao eixo do motor, é responsável por manter a circulação pressurizada e distribuir lubrificante a todas as peças móveis criando uma fina película que minimiza o contato entre as superfícies. Ao final do processo, o óleo retorna ao cárter por ação da gravidade, onde ele pode trocar com o meio exterior o calor absorvido das peças com as quais teve contato. Logo, observa-se também o papel de resfriamento (INFOMOTOR, 2009b, 2009c).

Figura 4 - Sistema de lubrificação.



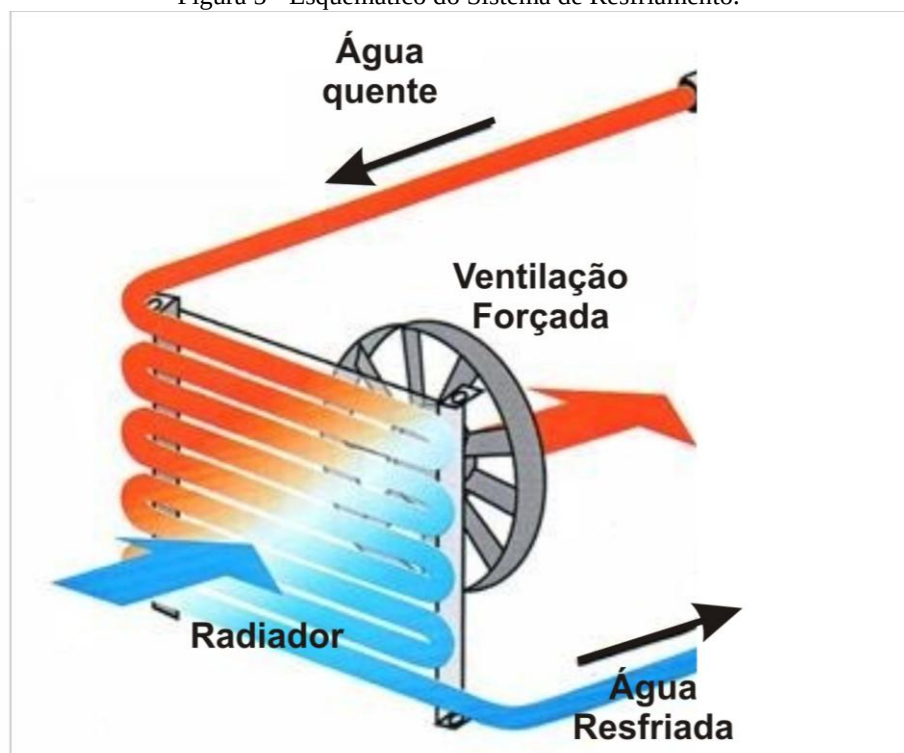
Fonte: <<http://ruralwillys.tripod.com/manutencao/pressaooleo.htm>>

- SISTEMA DE RESFRIAMENTO

O motor de combustão interna, em funcionamento, gera uma quantidade de calor muito grande. Segundo NETO (2012), na fase de compressão do ar, a temperatura do ar pode chegar a 900 °C. Logo, para manter a temperatura em níveis suportáveis se utiliza de um sistema de resfriamento composto por uma bomba d'água que força a circulação de água em um circuito fechado e pressurizado. Uma parte deste circuito fechado é denominado radiador, local no qual ocorre a troca de calor com o meio externo. Em algumas situações, pode-se ter o auxílio de uma ventoinha que força a passagem de fluxo de ar (Figura 5) (INFOMOTOR, 2009a).

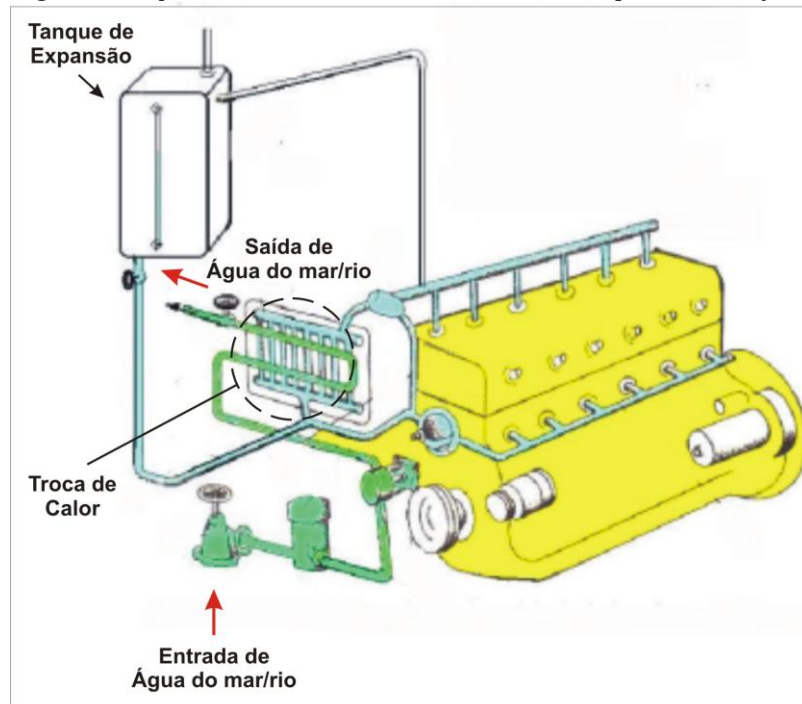
Em embarcações, pode-se ter o uso de uma bomba de água salgada que coleta a água através de uma válvula de fundo, envia para o trocador de calor e por fim descarrega através de uma válvula de descarga (Figura 6).

Figura 5 - Esquemático do Sistema de Resfriamento.



Fonte: <<http://www.forumbmwportugal.com/showthread.php?t=361>>
(Editada pelo próprio autor)

Figura 6 - Esquemático de um Sistema de Resfriamento para embarcações.



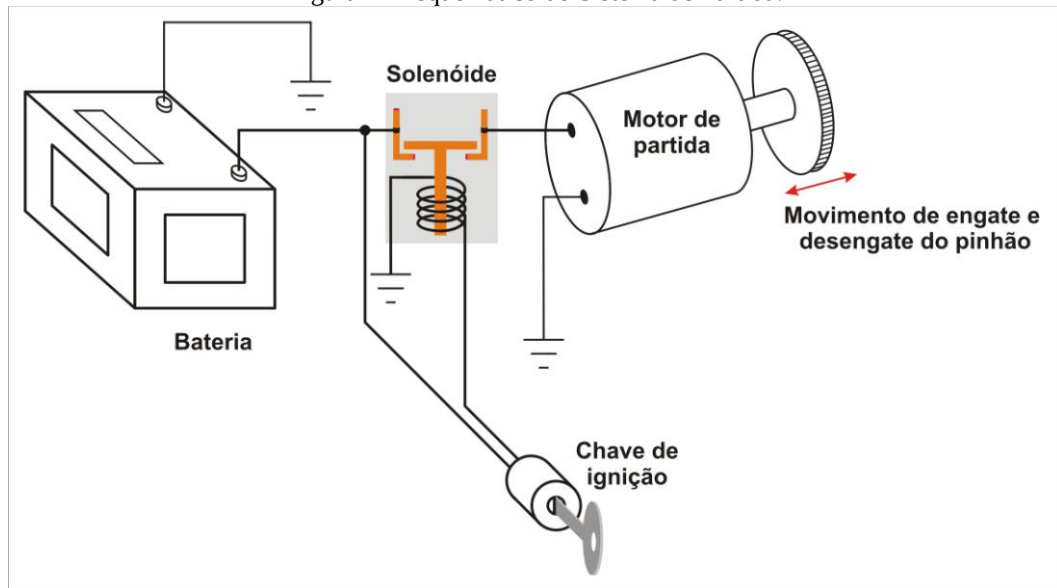
Fonte: <<http://www.ebah.com.br/content/ABAAAAgfsAF/maquinas-propulsao>>
(Editada pelo próprio autor)

- SISTEMA DE PARTIDA

O motor de combustão interna é incapaz de funcionar sem o auxílio do sistema de partida, também denominado arranque. Este sistema é responsável por garantir as primeiras voltas do eixo, ou seja, tirá-lo da inércia. O sistema de partida pode ser pneumático, elétrico, hidráulico, ou por injeção de ar comprimido nos cilindros. O mais usual é o sistema elétrico (MARINHA DO BRASIL, 2008, p. 168) composto de um motor geralmente ligado em 12 volts que no seu eixo possui uma engrenagem (pinhão do motor de arranque) que pode se movimentar axialmente podendo assim se acoplar a cremalheira na partida (Figura 3). Segundo Pereira (2006), a maioria dos motores possui uma cremalheira com 132 dentes e a relação cremalheira/pinhão pode chegar até 20:1.

Conforme pode ser visto na Figura 7, no sistema de partida existe uma chave de ignição, a qual, ao ser acionada, permite que o motor entre em funcionamento e o pinhão se desloque axialmente engrenando da cremalheira. No momento que o motor de combustão entra em funcionamento, deve-se soltar a chave elétrica e o pinhão deve ser automaticamente desacoplado da cremalheira.

Figura 7 - Esquemático do Sistema de Partida.

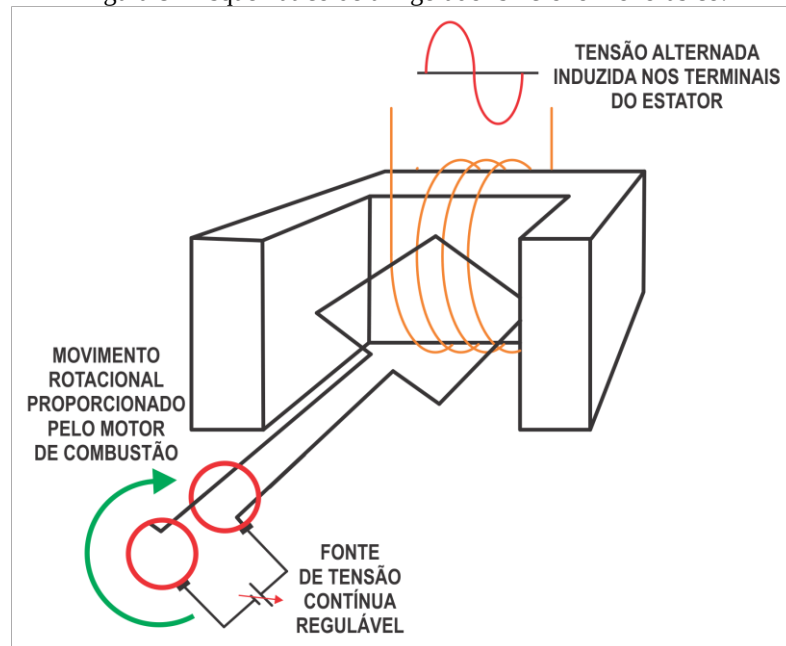


Fonte: Produção do próprio autor.

- SISTEMA DE GERAÇÃO DE ENERGIA ELÉTRICA

Para manter as baterias em boas condições e devidamente carregadas é inserido na instalação um carregador automático ou um gerador de carga, conhecido como alternador, que fica acoplado ao motor através de uma correia. Neste tipo de máquina, o enrolamento de campo é definido pelo rotor e está submetido a uma tensão contínua por meio de escovas estacionárias e anéis coletores. Uma força motriz faz com que o rotor gire e, com o movimento relativo entre o campo magnético constante do enrolamento do rotor e o do estator, é induzida tensão alternada nos terminais dos enrolamentos do estator (FITZGERALD, 2006). A Figura 8 ilustra o que foi explanado embora esteja representando um gerador monofásico e o alternador seja trifásico, por apresentar maior potência.

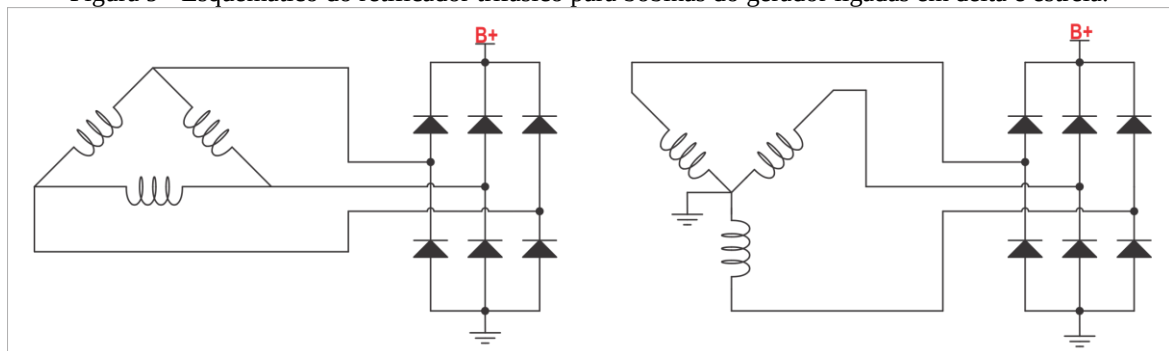
Figura 8 - Esquemático de um gerador síncrono monofásico.



Fonte: Produção do próprio autor.

Este alternador é um gerador síncrono que produz uma tensão alternada a qual, ao passar por um retificador, composto de diodos, como pode ser visto na Figura 9, é transformada em tensão contínua. Esta tensão contínua, disponível no terminal B+, é ligada ao terminal positivo da bateria.

Figura 9 - Esquemático do retificador trifásico para bobinas do gerador ligadas em delta e estrela.



Fonte: Produção do próprio autor.

O gerador é chamado de síncrono justamente porque a frequência, $f_{alternador}$, da tensão induzida é proporcional ao número de pólos $p_{alternador}$ da máquina e da velocidade de rotação n_{motor} em rpm do rotor:

$$f_{alternador} = \frac{p_{alternador}}{2} \times \frac{n_{motor}}{60} \quad (1)$$

Segundo BOSCH (2001), os alternadores podem ser classificados, quanto ao tipo, em compactos, monoblocos e *standard*. Os alternadores de construção compacta possuem enrolamento trifásico com 12 pólos enquanto os do tipo monobloco possuem 12 ou 16 pólos e os do tipo *standard* 4, 6 ou 14 pólos. Quanto a aplicação, o primeiro é utilizado para carros de passeios e motocicletas; o segundo para carros de passeio, tratores, ônibus, cavalos mecânicos e máquinas de construção; e o terceiro é utilizado em veículos especiais e navios.

Alguns alternadores possuem um terminal externo chamado de pino W o qual possui uma ligação com uma das bobinas do estator, ou seja, neste terminal pode ser vista a tensão alternada gerada em uma das bobinas. Com isto, pode-se descobrir a velocidade do motor monitorando o valor da frequência desta tensão alternada.

Não só a frequência da tensão alternada depende da velocidade do motor, a amplitude desta tensão alternada também possui relação com a velocidade assim como depende também da excitação fornecida ao rotor. Como a tensão de saída deve ter amplitude constante para que as baterias possam ser carregadas, existe um componente no sistema responsável justamente por variar a excitação contínua de modo a controlar a tensão fornecida para a bateria, o regulador de tensão. Na partida, a tensão contínua necessária para excitação do campo é disponibilizada pelas baterias e posteriormente passa a ocorrer autoexcitação, ou seja, a excitação do campo é feita pela própria tensão gerada pelo alternador (BRITO, 2006).

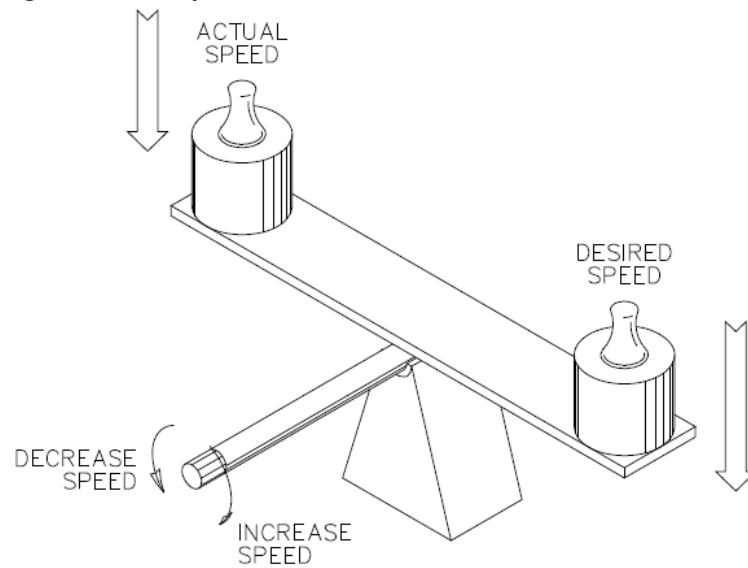
- SISTEMA DE COMBUSTÍVEL

O sistema de combustível é composto essencialmente pelo tanque de combustível, também chamado tanque de serviço, no qual é armazenado o combustível pronto para uso; pela bomba de transferência, também chamada de alimentadora justamente por ter a função de retirar combustível do tanque de serviço e o levar para a bomba injetora de combustível; e pelos injetores, uma composição da bomba injetora de combustível e da válvula de injeção, a qual abre e fecha permitindo a liberação ou não de combustível para os cilindros (MARINHA DO BRASIL, 2008).

A velocidade do motor de combustão depende tanto da carga quanto da quantidade de combustível injetada fazendo com que haja a necessidade de reguladores de velocidade, também chamados de governadores ou *governors*, que mantenham a rotação desejada independente da carga (controle isócrono). Estes dispositivos regulam o consumo de combustível, portanto, atuam na bomba injetora de combustível. Por exemplo, quando há um

decréscimo da carga, a rotação do motor aumenta e o governador atua de forma a fazer com que menos combustível seja liberado para os cilindros. Os governadores podem ser mecânicos, hidráulicos ou eletrônicos. Ambos possuem o mesmo princípio de funcionamento, o qual pode ser ilustrado pela Figura 10, mas os governadores eletrônicos vêm substituindo os hidro-mecânicos por possuírem um controle mais preciso.

Figura 10 - Balanço de velocidade referenciando controle de velocidade.



Fonte: WOODWARD (2011, p. 2).

3 NORMATIZAÇÕES

Como pode ser visto na seção anterior, o motor de combustão interna a diesel possui inúmeros sistemas e estes podem estar susceptíveis a falhas ou operações inadequadas as quais devem ser monitoradas para que não venham a comprometer a segurança da embarcação e dos tripulantes assim como a vida útil do próprio equipamento.

A IEEE 45 (2011, p. 58), a qual apresenta práticas recomendáveis para instalações elétricas em embarcações, dita, por exemplo, que o motor deve ser desligado sempre que ocorrer baixa pressão do óleo lubrificante, sobretemperatura da água de resfriamento e sobrevelocidade (*overspeed*) maior que 115% da velocidade nominal de operação. Operações de desligamento (*shutdown*) opcionais também são citadas na mesma norma com relação a ocorrência de baixo nível de óleo, baixo nível de água de resfriamento, baixa velocidade de rotação (*underspeed*) e sobretemperatura do óleo lubrificante.

A IEEE 45 (2011, p. 230) também faz exigências pontuais com relação aos sistemas lubrificante, arrefecedor e outros que envolvam fluidos. Define-se a necessidade de um sistema de alarme visível e audível na central de operação ou em locais com presença humana quando ocorrer um desligamento por falha em algum destes sistemas (perda de pressão do óleo lubrificante ou alta temperatura do água de resfriamento). Além disto, os dispositivos de monitoramento devem ser capazes de emitir sinais de alerta antes de tais valores de temperatura ou pressão serem atingidos.

Todas as normas técnicas desenvolvidas pelas sociedades classificadoras são todas baseadas na IEEE *Standard* 45 (2002). Um trecho de um dos guias da ABS pode ser lido abaixo:

Automatic Safety Shutdown to avert rapid deterioration of machinery, the following automatic shutdowns are to be provided, regardless of the mode of control: manual, remote or automatic. Automatic shutdowns are to be completely independent, including sensors, of the control and alarm systems. These shutdowns are not to be fitted with manual override. i) For all diesel engines: Overspeed. (ABS, 2003, p. 13)

A BV faz restrições quanto à potência das máquinas as quais se refere:

In addition to the speed governor, auxiliary engines of rated power equal to or greater than 220 kW driving electric generators are to be fitted with a separate overspeed protective device, with a means for manual tripping, adjusted so as to prevent the rated speed from being exceeded by more than 15%. This device is to automatically shut down the engine. (BV, 2007, p. 29)

4 SENSORES

Compreendidos os principais sistemas do motor de combustão interna e quais são as exigências normativas, adentra-se no estudo de como pode ser feito o monitoramento das variáveis exigidas (pressão, temperatura e velocidade). São utilizados sensores, analógicos ou digitais, para converter os valores destas grandezas físicas em grandezas elétricas (resistência, corrente ou tensão). A seguir, são apresentados exemplos de sensores com tais funções.

4.1 Sensor de temperatura

Os sensores de temperatura podem medir a temperatura do óleo de lubrificação ou da água de resfriamento. Alguns possuem semicondutores eletrônicos cuja resistência varia com a temperatura e são denominados termistores (*thermally sensitive resistor*). Como a resistência pode assumir qualquer valor ao longo do tempo, estes são classificados como sensores analógicos. Os termistores podem ser do tipo PTC, quando a resistência aumenta com a temperatura, e NTC, quando a resistência diminui com o aumento da temperatura.

Além dos termistores existem os termopares, as termorresistências (as quais possuem um filamento de um metal o qual a resistência varia com a temperatura, por exemplo, o Pt-100), o par-bimetálico, os sensores eletrônicos e os pirômetros, (THOMAZINI, 2007).

O par bimetalico é composto de duas chapas de materiais com diferentes índices de dilatação. Ao serem aquecidos, ocorre uma deformação nas chapas propiciando a abertura ou o fechamento de um contato para uma determinada temperatura. Também podem ser chamados de interruptores térmicos.

Na Figura 11, pode ser visto um exemplo de um sensor de temperatura analógico e na Figura 12, alguns exemplos de interruptores térmicos, ambos do fabricante VDO.

Figura 11 - Sensor de Temperatura (termistor).



Fonte: <www.vdo.com.br>

Figura 12 - Interruptores térmicos.



Fonte: <www.vdo.com.br>

4.2 Sensor de pressão

Os sensores de pressão que são utilizados para medir a pressão do óleo lubrificante geralmente possuem terminais elétricos nos quais pode ser lida a resistência de acordo com o valor da pressão e podem ter também um contato que é acionado por um diafragma de metal o qual se deforma proporcionando a abertura ou fechamento do contato a partir de um limiar de pressão (VDO, 2013).

Na Figura 13, é apresentado um sensor de pressão do fabricante VDO o qual faz leitura de pressão de 0 a 10 bar (apresentando valores de resistência de 10 Ω para 0 bar e 88 Ω para 4 bar) e fecha o contato de alarme quando a pressão se torna menor ou igual a 0,5 bar. A medição da resistência deve ser feita entre o pino *G* e a carcaça e o contato deve ser considerado entre o pino *WK* a carcaça do sensor que passa a estar aterrada quando o sensor é instalado.

Figura 13 - Sensor de pressão de óleo



Fonte: <www.vdo.com.br>

4.3 Sensor de velocidade

O sensor de velocidade mais frequentemente utilizado para aferir velocidade da unidade motriz é o *pickup* magnético (MPU) que se trata de um transdutor eletromagnético e deve ser instalado radialmente a cremalheira do volante.

Segundo WOODWARD (2011), estes sensores consistem de um ímã permanente e de uma bobina alojados dentro de um invólucro de material não magnético (aço inox ou plástico, por exemplo).

Quando o dente da cremalheira, que deve ser de material ferromagnético (ferro, por exemplo), passa pelo *pickup* magnético, a relutância do trajeto magnético visto pelo ímã se altera e, conseqüentemente, uma variação do fluxo magnético também ocorre. A lei de indução de Faraday (1831) afirma que a corrente induzida em um circuito fechado por um campo magnético é proporcional ao número de linhas do fluxo que atravessa a área envolvida do circuito, por unidade de tempo. Em outras palavras, uma tensão será induzida devido à variação do fluxo magnético no tempo. Esta tensão terá frequência, f_{pickup} , dada pela relação entre a velocidade de rotação, n_{motor} em rpm, do motor e o número de dentes, n_{teeth} , da cremalheira:

$$f_{pickup} = \frac{n_{motor}}{60} \times n_{teeth} \quad (2)$$

Na Figura 14, pode ser observado o *pickup* magnético do fabricante KVA modelo KS3002.

Figura 14 - *Pickup* magnético.

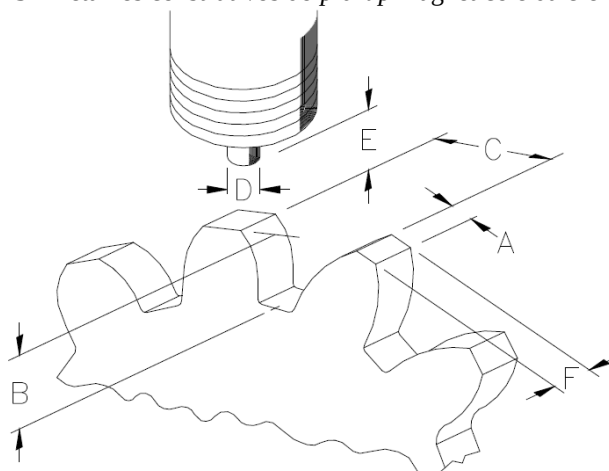


Fonte: < <http://kva.com.br/kva2.0/index.php/produtos/sensores> >

Alguns detalhes devem ser observados quanto à instalação deste sensor, quão próximo da cremalheira ele deve estar (observar Figura 15). A largura (representada pela dimensão A), a altura (representada pela dimensão B) e a espessura (representada pela dimensão F) dos dentes da cremalheira devem ser maiores que o diâmetro (dimensão D) do pólo do *pickup* magnético. O espaçamento entre o sensor e a cremalheira (denominado *gap*) deve ser de aproximadamente 0,3 mm, o que pode ser obtido enroscando o sensor até que ele toque nos dentes da cremalheira para então dar aproximadamente 3/4 de volta (para o sensor da Figura 14) no sentido contrário, anti-horário. E o espaçamento entre os dentes (dimensão C) deve ser maior ou igual a três unidades do diâmetro do sensor (WOODWARD, 2011).

Segundo Grupo Ayrestech (2013), estando o sensor instalado corretamente, a tensão fornecida nos seus terminais deve estar entre 1,7 e 13 Vca.

Figura 15 - Detalhes construtivos do *pickup* magnético e da cremalheira.



Fonte: WOODWARD (2011, p. 28).

Estando o sensor instalado corretamente, o sinal de tensão gerado pelo *pickup* pode, então, ser interpretado por um dispositivo e, visto que se sabe a relação entre a frequência desta forma de onda e a rotação através da Equação 2, a rotação do motor pode ser obtida.

Outra forma de se obter o valor da rotação do motor é fazendo-se uso do sinal de tensão gerado pelo alternador que fica disponível no pino W, como já foi explicado anteriormente, mas não se recomenda o uso deste sinal visto que pode haver imprecisão associada ao deslizamento da correia e, tão logo, o motor pode parar haja vista uma falha no alternador.

5 ESTUDO DE CASO

O objetivo do estudo de caso foi a instalação de um sistema que garanta as proteções mecânicas do MCA de um rebocador (Figura 16). Os dois grupos geradores (bombordo e boreste) são do fabricante Cummins, modelo C110 D6 4 (ver Figura 17), os quais possuem: potência nominal de 140 kVA em regime *standby* e 128 kVA em regime *prime*; motores de partida elétricos; alternadores de 12 V contínuos e 4 pólos; e são acionados por motores a diesel modelo 6BT 5.9-G6 com 6 cilindros, rotação nominal de 1800 rpm e consumo de 31 l/h (quando operando com 100% de carga em regime *standby*) ou 29 l/h (quando operando com 100% de carga em regime *prime*).

Figura 16 - Rebocador.



Fonte: Produção do próprio autor.

Figura 17 - Gerador modelo C110 D6 4 do fabricante Cummins.



Fonte: Cummins, 2011, p. 1.

O sistema instalado é composto do sensor de temperatura da água, do sensor de pressão de óleo, do sensor de velocidade (*pickup* magnético) e do controlador juntamente com o botão de emergência, o *giroflex* e a sirene.

5.1 Controlador

A empresa ComAp fabrica controladores de motores para diversas aplicações. Será alvo das apresentações seguintes um controlador, InteliDrive Lite (Figura 18), que foi instalado com o objetivo de viabilizar o monitoramento de variáveis como velocidade, pressão e temperatura e efetuar o *shutdown* automático para as situações exigidas pelas normas citadas anteriormente.

Figura 18 - Controlador InteliDrive Lite da ComAp.



Fonte: <<http://www.comap.cz/products/detail/new-intelidrive-lite/#lightbox/0/>>

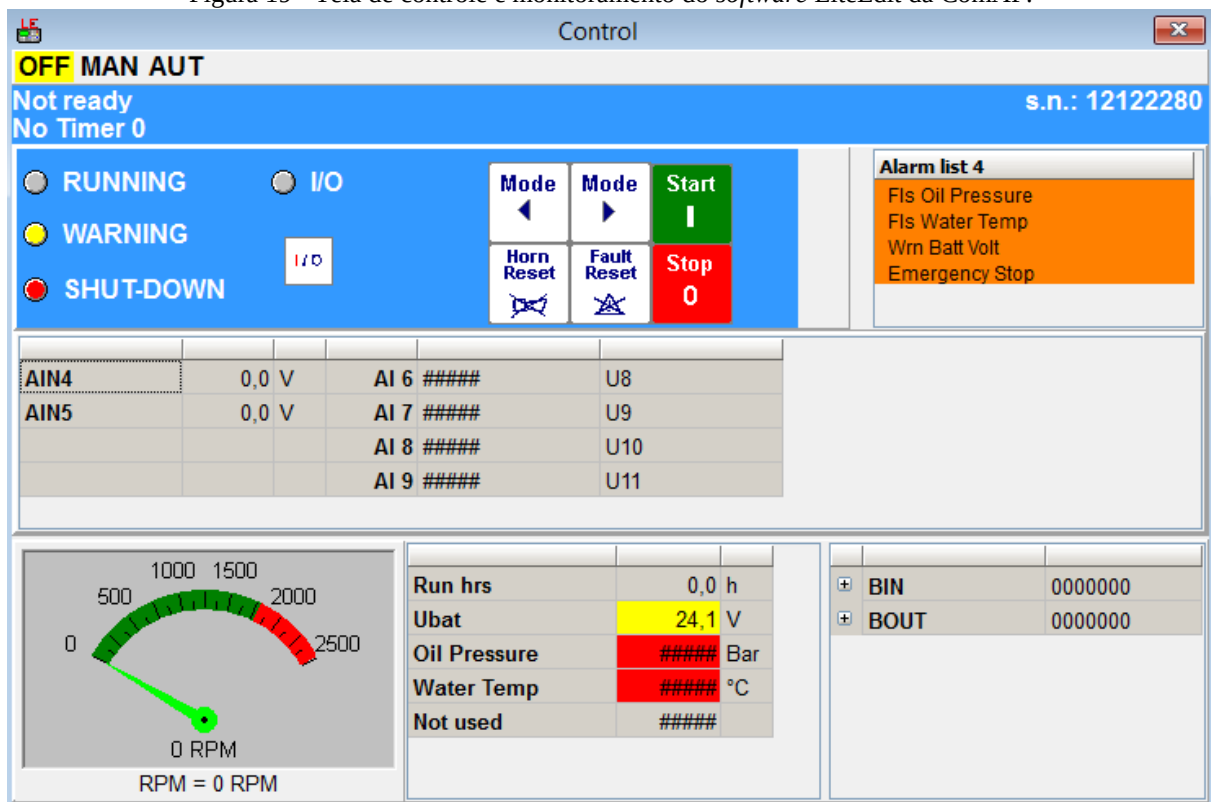
O controlador em estudo possui as seguintes características principais: tensão de alimentação contínua de 8 a 36 V; consumo entre 40 e 430 mA que dependem da tensão de alimentação e da temperatura; dimensões de 180x120x55 mm; 7 entradas e 7 saídas ambas digitais e configuráveis; 9 entradas analógicas; e a entrada do sensor de velocidade (*pickup*) o qual deve fornecer uma tensão mínima de 2 V pico a pico e máxima de 50 V eficazes, além de uma frequência entre 4 e 10 kHz (ComAp, 2012).

5.2 Configurações

É através do *software* LiteEdit, disponibilizado pela ComAp, que pode ser feito o acompanhamento em tempo real (caso o controlador esteja conectado via cabo serial ao computador) do *status* ou dos valores monitorados de todas as saídas e entradas do controlador, bem como o ajuste de todos os parâmetros e configurações.

Na Figura 19, pode ser vista a tela principal do *software* na qual podem ser acompanhados, em tempo real, o valor da velocidade (em rotações por minuto), o tempo de operação do motor, a tensão da bateria, a pressão do óleo, a temperatura da água, o *status* das 7 entradas e 7 saídas ambos binários (zero ou um) e a lista de alarmes. Também é possível comandar o controlador através dos botões *start*, *stop*, *fault reset* e *horn reset* que são uma reprodução da IHM do controlador.

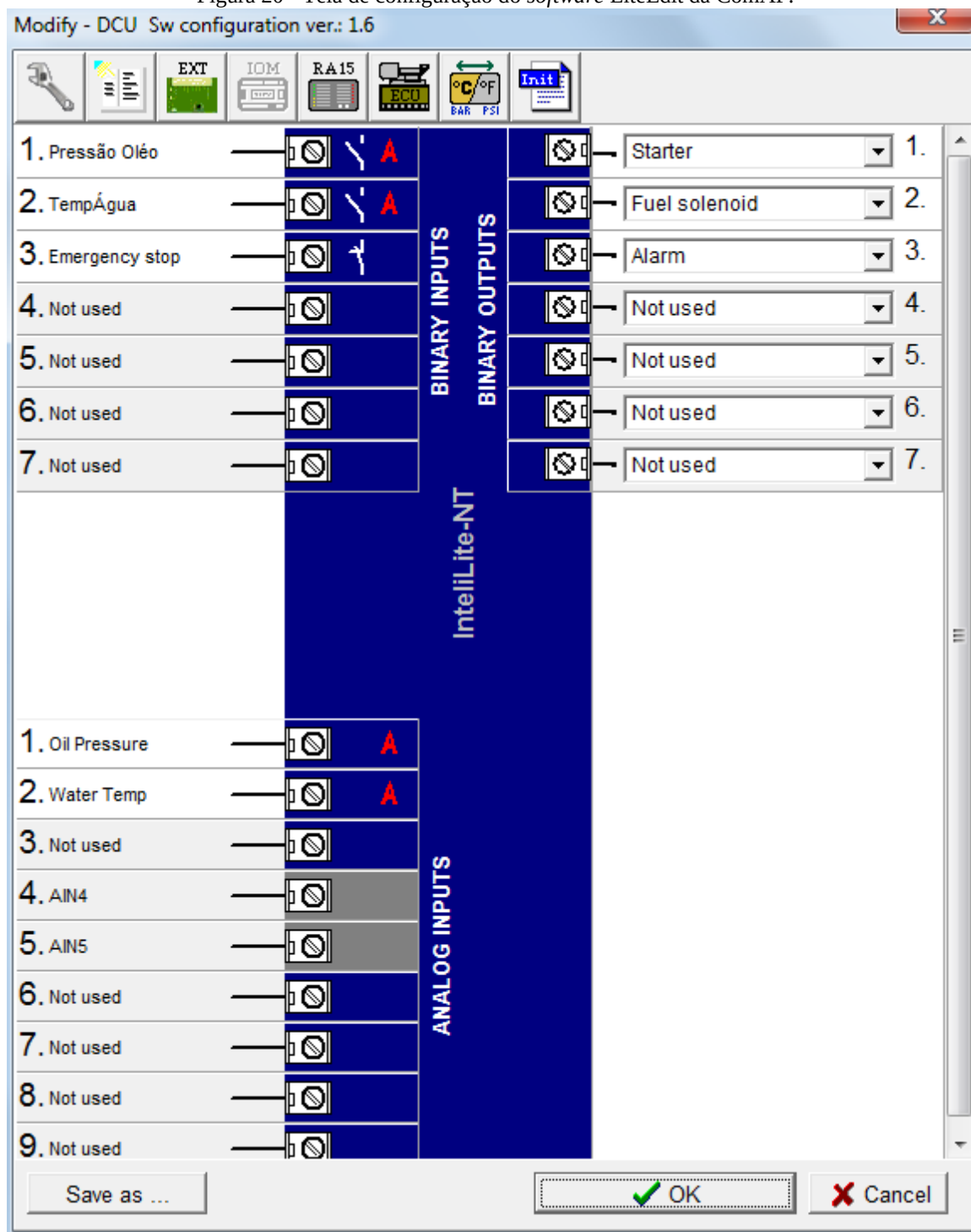
Figura 19 - Tela de controle e monitoramento do *software* LiteEdit da ComAP.



Fonte: Produção do próprio autor.

Na Figura 20, pode ser vista a tela de configuração das entradas e saídas do *software*. Nos parágrafos seguintes serão detalhadas as possíveis configurações destas entradas e saídas, assim como quais foram escolhidas e qual o princípio de funcionamento do sistema completo.

Figura 20 - Tela de configuração do software LiteEdit da ComAP.



Fonte: Produção do próprio autor.

5.2.1 Saídas digitais

Inúmeras podem ser as configurações das saídas digitais. Elas podem sinalizar (fechando um contato seco da saída) a ocorrência de uma falha de velocidade (*underspeed* ou *overspeed*); a ocorrência de um *shutdown* ou *warning* qualquer assim como os específicos (pressão do óleo, temperatura da água e nível de óleo); o modo no qual o motor está operando que pode ser desligado, manual, automático, ou seja, tem uma saída *OFF*, *MAN* e *AUT* que quando adotadas como saídas digitais fecham seus contatos se o modo estiver ativo; o *status* das entradas digitais; entre muitas outras opções (ComAp, 2013).

As principais saídas digitais são denominadas *starter* e *fuel solenoid*, pois são as que definem o funcionamento do sistema. A saída *starter* é a que possui relação com o sistema de partida do motor a diesel tendo a função da chave de ignição que permite o acionamento do motor de partida. A saída *fuel solenoid* está relacionada com o sistema de combustível: quando o contato desta saída fecha, a válvula de combustível deve ser aberta permitindo a passagem de combustível. Estas duas saídas digitais foram ativadas neste projeto.

O motor pode ser acionado manualmente, através dos botões *start* e *stop* presentes no painel do controlador, ou automaticamente, através de entradas digitais que tenham a mesma função dos botões, mas são comandadas por um controlador de velocidade ou por botões remotos.

Operando manualmente, ao ser pressionado o *start*, um *delay* de tempo denominado *prestart time* (configurável entre 0 e 500 s, mas adotado de 2 s) decorre até que a saída *starter* seja acionada ligando o motor de partida. Um segundo (1 s) antes da saída *starter* ser acionada, automaticamente a saída *fuel solenoid* é acionada e é liberado o combustível (diesel) necessário ao funcionamento do motor. O contato da saída *start* continua fechado até que o motor consiga atingir uma velocidade configurável e denominada *starting rpm*, que deve ser um valor suficiente para que o motor saia da inércia. Para esta aplicação foi adotado um valor de 350 rpm.

Um outro parâmetro que deve ser configurado é o chamado *maxcrank time* (configurável entre 1 e 60 s, mas adotado de 5 s), que significa o tempo máximo no qual o motor de partida fica ligado tentando fazer com que o motor diesel tente atingir a velocidade de 350 rpm. Superado este tempo, se o motor diesel não atinge a velocidade de 350 rpm: um sinal de *shutdown* é gerado fazendo com que o contato *start* e *fuel solenoid* sejam

imediatamente abertos interrompendo a partida; um sinal visual de erro de partida (*start fail*) é gerado; e o controlador muda do *status starting* para *shutdown*. Não ocorrendo nenhuma falha (*maxcrank time* atingido, nível de óleo baixo, falha na abertura da válvula de combustível, baixa pressão de óleo, perda do sinal do sensor de velocidade, dentre outras), ao atingir esta velocidade (*starting rpm*): a saída *start* é aberta; a saída *fuel solenoid* continua fechada; e o controlador muda do *status starting* para *running*.

Estando o motor operando em suas condições normais (*status running*), ao ocorrer alguma falha que gere o comando de *shutdown*, imediatamente a saída *fuel solenoid* é aberta e o motor para. Porém se o motor está operando normalmente (*status running*) e deseja-se pará-lo, deve-se pressionar o botão *stop* e só depois de um certo tempo denominado *cooling time* (configurável entre 0 e 3600 s, mas adotado de 30 s) é que o motor para definitivamente, ou seja, a saída *fuel solenoid* é aberta.

O tempo de resfriamento (*cooling time*) é o tempo no qual o motor opera sem carga antes de desligar. Este tempo é utilizado quando se utiliza alguma saída digital do controlador (*cooling* ou *ready to load*) para operar na abertura do disjuntor que interliga o gerador ao barramento nos quais estão as cargas elétricas. Mesmo estando configurado o tempo de resfriamento, este pode ser ignorado pressionando-se o botão *stop* duas vezes.

Existem outras duas saídas digitais, *alarm* e *horn*, que podem ser usadas e que dizem respeito aos alarmes visuais e sonoros (exigência normativa) visto que geralmente são utilizadas para acionar um giroflex e uma sirene. Ambas são acionadas quando ocorre um sinal de *warning* ou *shutdown*. Após o contato da saída *alarm* ser fechado, este só é aberto ao ser pressionado o botão *fault reset* presente no painel do controlador (Figura 21), que também reinicializa os sinais de falha permitindo, caso seja necessário, nova partida do motor. Após o contato da saída *horn* ser fechada, só se consegue abrir pressionando o botão *horn reset* também presente no painel do controlador (Figura 22).

No projeto proposto, optou-se por ativar apenas a saída *alarm* e utilizá-la para acionar, simultaneamente, o giroflex e a sirene. Ocorrida uma falha, caso se deseje desligar o giroflex e a sirene, pressiona-se o botão *fault reset*. Como este botão também reinicializa os sinais das falhas, estas podem ser acompanhadas na aba de histórico de falhas que está presente na IHM (Interface Homem-Máquina) do controlador.

Figura 21 - Botão *fault reset* do controlador InteliDrive Lite da ComAp.



Fonte: ComAp, 2013, p. 46.

Figura 22 - Botão *horn reset* do controlador InteliDrive Lite da ComAp.



Fonte: ComAp, 2013, p. 46.

5.2.2 Entradas digitais

As entradas digitais podem ser configuradas como sendo de proteção (*alarm*) ou de controle (*control*). Se for determinada a opção de controle, deve ser escolhida qual é a função desta entrada dentre as opções fornecidas: entrada remota para *start* ou *stop*, parada de emergência (*emergency stop*), entrada remota para desligamento, dentre outras. E para todas as entradas, pode ser definido qual seu estado normal de operação (normalmente aberto ou fechado). Neste projeto, apenas a parada de emergência foi configurada como uma entrada digital de controle e foi utilizado um botão normalmente fechado.

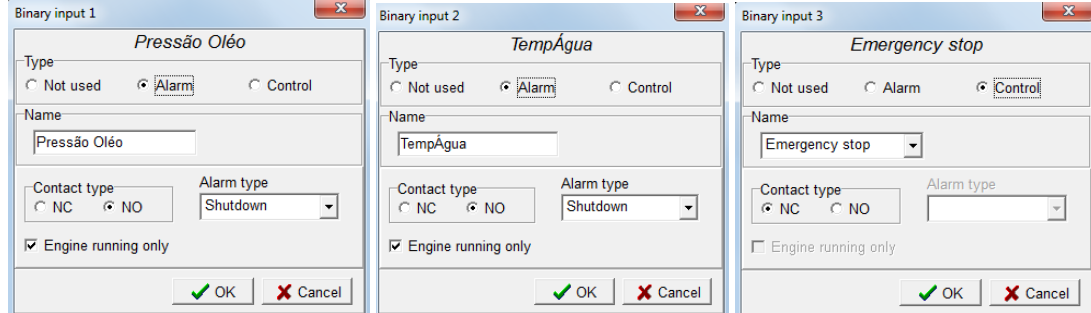
Definida uma entrada digital como sendo de controle, pode ser estabelecido pelo usuário o nome da entrada e o seu estado padrão. Para este projeto, foram definidas duas entradas de controle, uma para a temperatura da água (*TempÁgua*) e outra para pressão do óleo (*Pressão Óleo*), ambas normalmente abertas que vão, quando mudarem de estado, proporcionar o *shutdown* do motor.

O sensor de pressão utilizado neste projeto apresenta continuidade entre o pino WK e a carcaça quando a pressão do óleo se torna igual ou inferior a 0,5 bar e o interruptor térmico apresenta continuidade entre seus dois contatos a partir de 100 °C.

Na Figura 23 pode ser vista a tela de configuração de cada uma das três entradas digitais (*binary input 1*, *binary input 2* e *binary input 3*) citadas anteriormente (*Pressão Óleo*, *TempÁgua* e *emergency stop*). Um detalhe importante deve ser observado quanto ao sensor de pressão: quando o motor está desligado, não há pressão no óleo lubrificante visto que a bomba está desligada, logo, entre o pino WK e a carcaça existe continuidade e assim o motor não consegue ser acionado, pois esta entrada digital sempre estaria gerando sinal de alarme. Para

contornar tal situação, deve ser feita uma configuração de forma que esta entrada digital só seja verificada caso o motor esteja em funcionamento. Como pode ser visto na Figura 23, foi habilitada a opção *engine running only*.

Figura 23 - Tela de configuração das entradas digitais no *software* LiteEdit da ComAP.



Fonte: Produção do próprio autor.

5.2.3 Entradas analógicas

O controlador em estudo possui capacidade de fazer leitura de resistência entre 0 e 2,4 $k\Omega$ e neste projeto foram utilizados dois sensores resistivos, um de temperatura e um de pressão ambos configurados como de controle, como pode ser visto na Figura 24, que apresenta as telas de configurações das duas entradas analógicas (*analog input 1* e *analog input 2*) intituladas, respectivamente, *Oil Pressure* e *Water Temp*.

Para que o controlador saiba qual a relação existente entre o valor da resistência lida e a variável real (pressão em bar ou temperatura em Graus Celsius) deve ser configurada a curva do sensor a qual pode ser obtida por meio de ensaios com os sensores. Na Figura 25, pode ser vista a curva do sensor de pressão e a na Figura 26 a curva do sensor de temperatura, ambas definidas pelo usuário para os sensores que foram instalados na carcaça do motor.

As duas entradas analógicas foram configuradas para gerar sinal de *warning* e *shutdown*, como pode ser visto na definição do tipo do alarme na Figura 24 (*type: Wrn+Sd*) e os valores a partir dos quais serão gerados estes sinais são definidos na tela de configurações, nas variáveis: *Oil Pressu Wrn*, *Oil Pressu Sd*, *Water Temp Wrn* e *Water Temp Sd* que foram definidas, respectivamente, 2,5 bar, 2 bar, 92 °C e 98 °C. Por exemplo, caso a pressão do óleo atinja o valor de 2,5 bar, apenas um sinal de alarme será gerado. Mas, se a pressão continuar diminuindo, quando for alcançado o valor de 2 bar, imediatamente o motor será desligado.

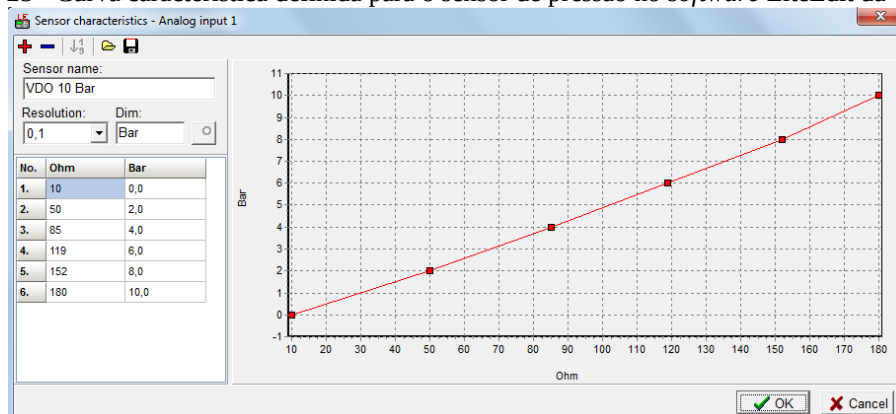
O sistema de proteção passa a ser redundante visto que são gerados sinais de *shutdown* pela entrada digital de pressão e temperatura assim como através das entradas

analógicas. O motivo desta redundância é justamente fazer com que o sinal de *shutdown* não deixe de ser gerado caso algum sensor deixe de operar. Por exemplo, o sinal de *shutdown* deve ser gerado a partir de 2 bar, mas caso o sensor resistivo apresente defeito, o sensor digital vai fechar seus contatos para uma pressão inferior ou igual a 0,5 bar e o sinal de *shutdown* será gerado. Da mesma forma para proteção de sobreaquecimento: o sensor resistivo deve gerar *shutdown* assim que a temperatura atingir o valor de 98 °C, mas caso este sensor falhe, o interruptor térmico será responsável por garantir o *shutdown* assim que a temperatura da água atingir 100 °C.

Figura 24 - Tela de configuração das entradas analógicas no *software* LiteEdit da ComAP.

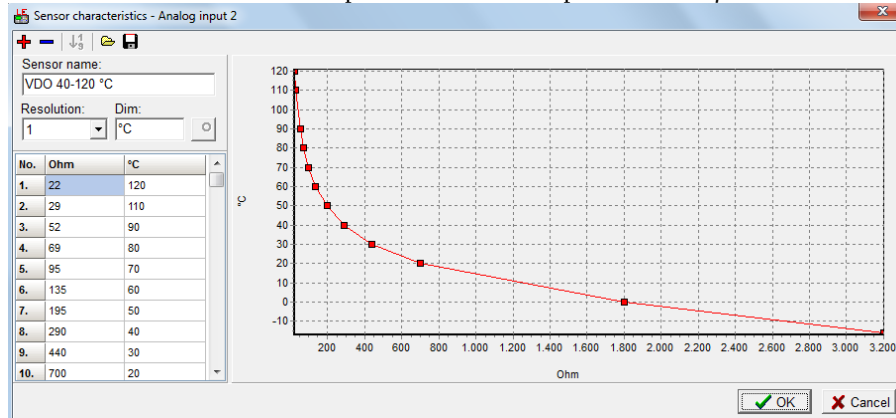
Fonte: Produção do próprio autor.

Figura 25 - Curva característica definida para o sensor de pressão no *software* LiteEdit da ComAP.



Fonte: Produção do próprio autor.

Figura 26 - Curva característica definida para o sensor de temperatura no *software* LiteEdit da ComAP.



Fonte: Produção do próprio autor.

5.2.4 Sensor de velocidade

É através dos pinos *RPM IN* e *RPM GND* que o sinal do sensor de velocidade deve ser fornecido ao controlador. O sensor deve ser um *pickup* magnético e para ser feita a leitura correta da velocidade é necessário ajustar o parâmetro *Gear Teeth* que indicará ao controlador quantos dentes possui a cremalheira. O controlador faz a leitura da frequência da forma de onda advinda do *pickup*, f_{pickup} e, com o número de dentes, n_{teeth} , é obtida a velocidade do motor, n_{motor} , de acordo com a Equação 2 apresentada na Seção 4.3. O valor de *Gear Teeth* pode ser ajustado de 0 a 500, mas caso seja ajustado igual a zero, o controlador ignora a leitura de velocidade do motor.

Para que a proteção de sobrevelocidade atue, outras variáveis têm de ser definidas. São elas: *Nominal RPM* (ajustado para 1800rpm) e *Overspeed* (configurável de 50 a 150% e ajustado para 110%).

Uma outra variável, intitulada *RPMbyWterminal*, cujo valor padrão é unitário, mas pode ser ajustada entre 0,5 e 2 com passo de 0,01, deve ser utilizada caso não seja utilizado o *pickup* para leitura de velocidade e sim o pino *W* do alternador. Trata-se de multiplicador que viabiliza um ajuste fino da velocidade fornecida pelo pino *W* do alternador.

Neste projeto foram feitos testes tanto com o *pickup* magnético tanto com o pino *W* do alternador. Os valores ajustados para *Gear Teeth* e *RPMbyWterminal* foram obtidos experiente verificando-se a frequência no barramento do gerador. Estando o valor da frequência em 60 Hz, sabe-se que o motor está operando com 1800 rpm então ajustou-se, utilizando o *pickup* magnético, o *Gear Teeth* com valor 152 e o *RPMbyWterminal* com valor

unitário. Utilizando o pino *W* do alternador, a variável *Gear Teeth* passou a ter valor de 32 e a variável *RPMbyWterminal* o valor de 1,95.

5.3 O projeto utilizando o controlador InteliDrive Lite da ComAp

Explicadas e definidas todas as entradas digitais e analógicas e saídas digitais, foi feita a instalação e ligação do controlador aos sensores (temperatura, pressão e velocidade) e aos relés que são acionados pelas saídas digitais do controlador.

Um resumo das saídas digitais utilizadas pode ser visto na Tabela 1; das entradas digitais na Tabela 2; das entradas analógicas na Tabela 3; e das principais variáveis as quais foram configuradas na Tabela 4, ambas presentes no APÊNDICE A.

O esquemático das ligações pode ser visto na Figura 39 (APÊNDICE A) no qual existem os relés denominados K1, K2 e K3 para acionar, respectivamente, o solenóide combustível, a chave de ignição do sistema de partida elétrica e, simultaneamente, o giroflex e a sirene do sistema de alarme.

Os detalhes da instalação podem ser visualizados nas Figura 40, Figura 41, Figura 42, Figura 43 e Figura 44 do APÊNDICE B.

5.4 Ensaaios

Os ensaios foram feitos simulando-se as ocorrências de sobretemperatura da água, perda da pressão do óleo, *overspeed* e parada de emergência.

De acordo com a curva configurada para a entrada analógica de temperatura, quanto menor a resistência medida, maior o valor interpretado da temperatura em graus Celsius. Logo, foi feita uma ligação entre os terminais do sensor de temperatura fazendo com que a resistência lida pelo controlador seja apenas a do cabo. Consequentemente, um sinal de *shutdown* foi gerado.

De acordo com a curva configurada para a entrada analógica de pressão, ao contrário da de temperatura, quanto menor o valor da resistência lida, menor é o valor de pressão interpretado, mas a simulação pode ser feita da mesma forma para que o *shutdown* por baixa pressão do óleo seja gerado. Como, para o sensor de pressão, a leitura da resistência é feita no pino *G*, a simulação foi feita aterrando-se este pino. Na Figura 45, presente no

APÊNDICE B, podem ser examinadas as listas de alarmes apresentadas na IHM do controlador para os dois ensaios citados (elevada temperatura e baixa pressão).

Para realizar o teste de sobrevelocidade, a técnica utilizada foi a de alteração do valor da variável *Overspeed* depois que o motor já estiver funcionando. O controlador está sempre verificando o valor da velocidade do motor e no momento que esta velocidade ultrapassar o valor da multiplicação das variáveis *Overspeed* e *Nominal RPM*, o sinal de *shutdown* é gerado e o motor desligado. O valor de *Overspeed* para este projeto foi denominado 110% mas, no momento da simulação ele foi alterado, já estando o motor em funcionamento, para 80%, ou seja, o valor da velocidade lida, 1800 rpm, passou a ter valor superior ao limite estabelecido ($0,8 \times 1800 \text{ rpm} = 1440 \text{ rpm}$) e o sinal de *shutdown* foi gerado.

Por fim, foi simulada uma situação de emergência, ou seja, foi pressionado o botão de parada de emergência. Como já foi discutido anteriormente, este botão aciona uma entrada digital do controlador o qual, ao verificar mudança de estado desta entrada, comanda a parada (*shutdown*) do motor. O botão foi pressionado e o controlador operou de forma esperada. As listas de alarmes apresentadas na IHM para a simulação destas falhas (*overspeed* e situação de emergência) podem ser reconhecidas na Figura 46 (APÊNDICE B).

A Figura 47, também presente no APÊNDICE B, apresenta duas telas capturadas da IHM do controlador nas quais podem ser vistos os valores das variáveis monitoradas pelo controlador estando o motor operando em condições normais: a tensão da bateria é de 13,3 V; a pressão do óleo é de 4,8 bar; a temperatura da água é de 35 °C; e a velocidade do motor é de 1809 rpm.

6 PROJETO DE UM DISPOSITIVO DE PROTEÇÃO CONTRA *OVERSPEED*

Conforme foi exposto anteriormente, controladores podem ser instalados de forma a garantir todas as proteções necessárias e exigidas para os motores a diesel, mas, diante do custo de tais equipamentos, um circuito de comando pode ser projetado tendo o mesmo princípio de funcionamento e possuindo custo significativamente menor.

Para que este circuito de comando possa ser desenvolvido, são necessários dispositivos de proteção isolados de pressão, temperatura e velocidade. No quarto capítulo, foram apresentados exemplos de dispositivos com contatos que fecham para baixa pressão e alta temperatura e sabe-se que tais sensores são facilmente encontrados no mercado local. Quanto ao dispositivo com princípio de funcionamento simples e de baixo custo que feche ou abra contatos quando se tem uma ocorrência de *overspeed*, sabe-se que este não é facilmente adquirido, o que acaba forçando a utilização de um controlador como o citado anteriormente ou fazendo com que muitas embarcações operem sem esta proteção.

Então, foi neste contexto que se encontrou incentivo para iniciar uma pesquisa que viabilizasse a produção de tal dispositivo de proteção.

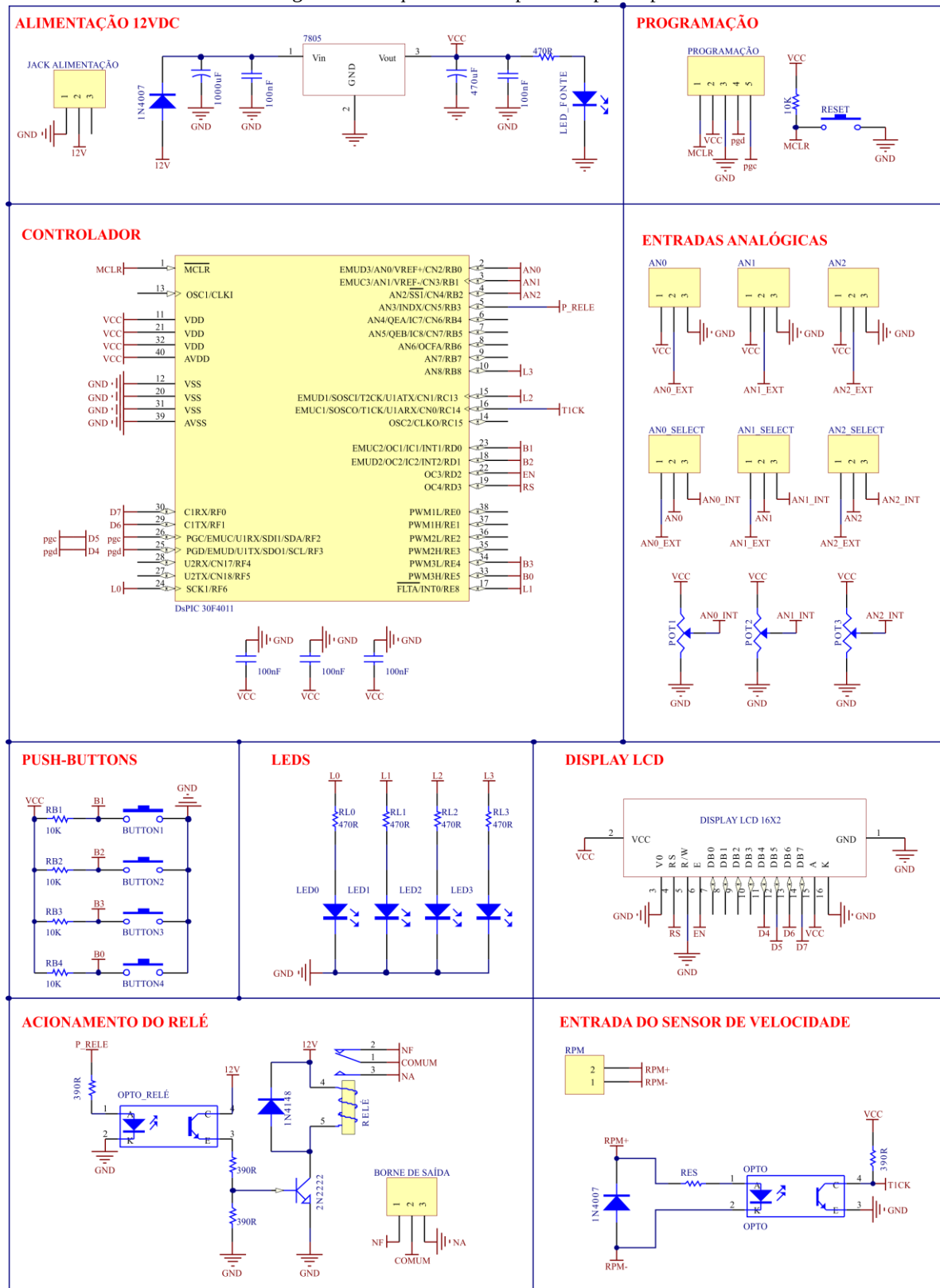
6.1 Implementação do primeiro protótipo

Um controlador digital de sinais, dsPIC, do fabricante Microchip foi escolhido para executar tarefas como leitura de sinais analógicos e digitais, controle do *status* de saídas digitais e execução de cálculos matemáticos.

Inicialmente foi escolhido o dsPIC30F4011 o qual possui 40 pinos, dentre os quais, 30 podem ser utilizados como entrada ou saída e foi desenvolvida uma placa com o maior número de acessórios possíveis (LEDs, botões, *display* LCD, relé, potenciômetros para simular sinais analógicos, entradas com optoacopladores, bornes e barramentos incluindo o de programação do dsPIC) visto que não se sabia quais necessidades iam aparecendo no decorrer do desenvolvimento do dispositivo.

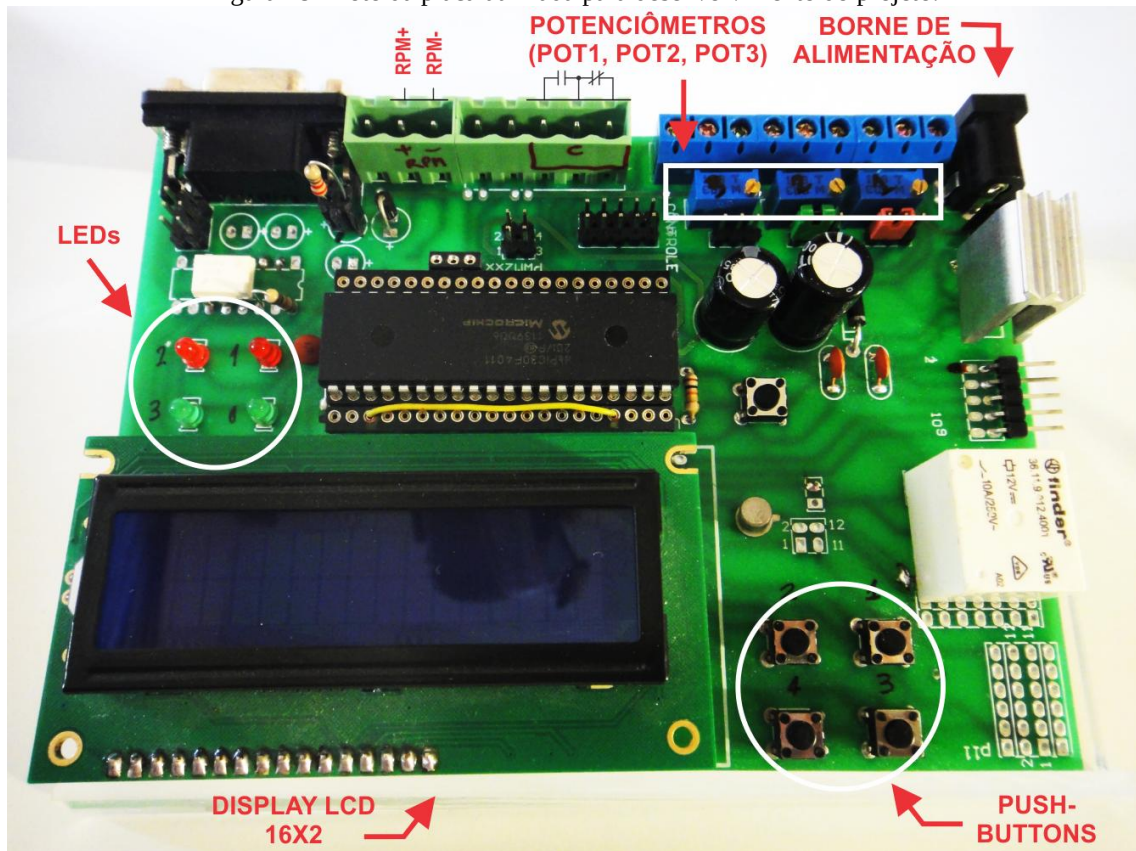
Na Figura 27 pode ser visto o esquemático da placa desenhado no software *Altium Designer* e na Figura 28 uma imagem da placa produzida.

Figura 27 - Esquemático do primeiro protótipo.



Fonte: Produção do próprio autor.

Figura 28 - Foto da placa utilizada para desenvolvimento do projeto.



Fonte: Produção do próprio autor.

6.1.1 Resumo dos componentes

Uma lista dos componentes que compõe a placa pode ser ligeiramente apresentada:

- Regulador de tensão que transforma a tensão de alimentação da placa, 12 V contínuos, em 5 V contínuos que irão alimentar o dsPIC;
- Barramento de programação através do qual pode ser feita a ligação entre os pinos *PGD*, *PGC*, *VSS*, *VDD* e *MCLR* do programador e do dsPIC;
- Circuito de *reset* do controlador com resistor de *pull-up* e um botão que, quando pressionado, provoca o aterramento da entrada *MCLR* gerando o *reset*;
- Circuito de acionamento do relé no qual foram utilizados um relé cuja bobina é acionada com 12 V contínuos, consome 30 mA e possui um contato reversível (1NAF); um optoacoplador para garantir a isolamento entre o relé e o dsPIC; um diodo de "roda livre" o qual tem função de oferecer um caminho alternativo para a corrente circular quando o

circuito indutivo da bobina for aberto; e um transistor que tem função de amplificar a corrente fornecida pelo pino do dsPIC;

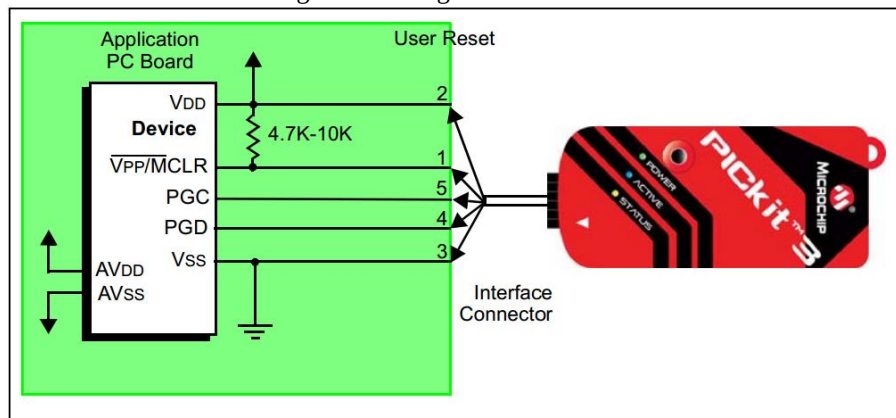
- Circuito que condiciona o sinal senoidal do sensor de velocidade composto de um optoacoplador o qual fará com que os sinais fornecidos ao dsPIC tenham valor nulo ou igual a 5 V contínuos já que na sua saída existe um resistor de *pull-up* ligado a malha de 5 V contínuos. Para evitar que qualquer tensão reversa seja fornecida ao optoacoplador (para o optoacoplador PC817 do fabricante SHARP, a máxima tensão reversa suportável é de 6 V), foi inserido um diodo na entrada o qual forçará a tensão reversa ser de, no máximo, 0,7 V o que ocorre quando a tensão entre os pinos RPM+ e RPM- possui valores negativos, polarizando o diodo, mas não acionando o LED interno do optoacoplador. Além do diodo, foi inserido um resistor para limitar a tensão (1,4 V contínuo) e a corrente (mínima de 5 mA e máxima de 1A) fornecida ao LED que aciona o foto-transistor;
- Três Potenciômetros (*POT1*, *POT2* e *POT3*) que fornecem uma tensão analógica de acordo com a posição do seu cursor rotativo. Estas tensões são fornecidas aos pinos configurados como entradas analógicas que estão ligados internamente ao conversor analógico-digital;
- Quatro Botões de impulso (*BUTTON1*, *BUTTON2*, *BUTTON3* e *BUTTON4*) com resistores de *pull-up* que irão garantir que a tensão fornecida ao dsPIC seja de 5 V quando não estejam pressionados e 0 V ao serem pressionados. Dois destes botões foram ligados aos pinos *INT1* e *INT2* configurados para detectar qualquer variação externa de nível de tensão, ou seja, quando o botão é pressionado, uma interrupção é gerada no código. Os outros dois foram ligados aos pinos *RE4* e *RE5*;
- Quatro LEDs (*LED0*, *LED1*, *LED2*, *LED3*) os quais tem o *status* controlado pelos pinos RF6, RE8, RC13 e RB8;
- *Display* LCD 16X2 (dezesseis colunas e duas linhas) o qual tem finalidade de expor os valores das variáveis desejadas.

6.1.2 Códigos

Todos os códigos apresentados neste projeto, foram desenvolvidos com o auxílio dos softwares MPLAB IDE versão 8.46 e o C30, ambos da Microchip, a mesma fabricante do dsPIC. O MPLAB é o ambiente de programação, mas toda vez que o código é compilado, o C30 é chamado internamente pelo MPLAB, ou seja, o C30 é o compilador.

Para carregar os códigos desenvolvidos dentro do dsPIC, foi utilizado o programador da Microchip PICKit 3 (Figura 29), que possui integração com o MPLAB, ou seja, estando o programador conectado a uma porta USB do computador e aos pinos de programação do dsPIC, a programação pode ser feita no mesmo ambiente de programação do MPLAB.

Figura 29 - Programador PICKit 3.



Fonte: MICROCHIP, 2009.

O dsPIC utiliza 4 pulsos do *clock* principal do sistema para gerar 1 ciclo de máquina. A fonte de *clock* utilizada foi de 7,37 MHz, fornecida pelo cristal interno do dsPIC, e foi configurado um multiplicador de 4. Então, a velocidade de processamento, que dita quantas milhões de instruções por segundo (MIPS) o processador é capaz de executar, é dada por:

$$F_{cy} = \frac{F_{osc} \times PLL}{4} = \frac{7,37 \text{ MHz} \times 4}{4} = 7,37 \text{ MHz} = 7,37 \text{ MIPS} \quad (3)$$

E o ciclo de máquina (CM), ou período T_{cy} do *clock* da máquina, tempo que o dsPIC leva para executar uma instrução, é dado por:

$$CM = T_{cy} = \frac{1}{F_{cy}} = \frac{1}{7,37 \text{ MHz}} = 135,68 \text{ ns} \quad (4)$$

Os principais periféricos utilizados foram os *timers* (*Timer 1*, *Timer 2* e *Timer 3*), os quais são contadores de borda de subida de um sinal que pode ser o próprio *clock* do

sistema (F_{cy}) ou qualquer sinal externo que pode estar sincronizado ou não com o *clock* do sistema. Cada *timer* é composto por três registradores de 16 bits: $TMRx$ que possui seu valor incrementado em uma unidade a cada borda de subida do sinal; o PRx que limita a contagem do $TMRx$ e chama uma interrupção sempre que o valor de $TMRx$ atinge o valor de PRx ; e o $TxCON$, um registrador de configuração a partir do qual pode ser definido, por exemplo, o *status* do *timer*, o valor do *prescale*, a fonte do sinal (interna ou externa através do pino $TxCK$) e, caso o sinal seja externo, se este é sincronizado ou não com o *clock* interno.

O *timer* 2 foi configurado para executar um bloco de comandos a cada 500 ms, denominado período do *timer*, T_{timer2} . Para tal, ele foi configurado para operar contanto bordas de subida do *clock* interno e com *prescale* de 256. O valor do registrador $PR2$ foi calculado e definido no código:

$$PR2 = \frac{T_{timer2}}{T_{cy} \times prescale} = \frac{500ms}{135,68 \text{ ns} \times 256} = 14395 \quad (5)$$

O *timer* 1 foi configurado para contar as bordas de subida do sinal gerado pelo optoacoplador que reflete o sinal gerado pelo sensor de velocidade. A cada borda de subida, o valor de $TMR1$ é incrementado, mas ao registrador $PR1$ foi atribuído valor máximo (65536, máximo valor para uma variável de 16 bits) justamente para que não sejam gerados estouros ou chamadas de interrupção. O valor deste registrador, $TMR1$, é lido a cada interrupção do *timer* 2, é armazenado na variável *CONTAGEM* e é zerado para que a contagem se reinicie. O valor armazenado nesta variável é exatamente metade do valor da frequência gerada pelo *pickup*:

$$CONTAGEM = TMR1 = \frac{f_{pickup}}{2} \quad (6)$$

O conversor analógico digital do dsPIC também foi utilizado para que o usuário, através de dois potenciômetros, possa definir dois parâmetros do motor: a velocidade nominal em rpm que será armazenada na variável V_n ; e uma constante definida como M , que depende do valor de dentes da cremalheira:

$$M = \frac{120}{n_{teeth}} \quad (7)$$

Foram configuradas, portanto, duas entradas analógicas, as quais estão nos pinos *AN0* e *AN1* que capturam o valor apresentado pelos potenciômetros 1 e 2 (*POT1* e *POT2*). O valor capturado é armazenado nos *buffers ADCBUF1* e *ADCBUF2* do conversor do dsPIC e transformados, respectivamente, nos valores da velocidade nominal *Vn* e no múltiplo *M* de forma que o valor da velocidade nominal pode ser configurado entre os valores de 1000 e 2023 rpm com passo de 10 rpm e o valor do múltiplo *M* pode ser configurado entre os valores de 0,6 e 2,4 com passo de 0,1.

Com o valor do múltiplo *M*, pode ser calculado o valor da velocidade do motor, n_{motor} , a cada 500 ms. Basta ser feita uma manipulação das relações das variáveis já apresentadas para descobrir qual relação existe entre eles:

$$M = \frac{120}{n_{teeth}} = \frac{120}{\left(\frac{60 \times f_{pickup}}{n_{motor}} \right)} = \frac{2 \times n_{motor}}{f_{pickup}} = \frac{2 \times n_{motor}}{(2 \times CONTAGEM)} = \frac{n_{motor}}{CONTAGEM} \quad (8)$$

Logo, para que seja obtido o valor da velocidade do motor a cada 500 ms, quando a interrupção do *timer 2* é chamada, o seguinte cálculo deve ser efetuado pelo dsPIC:

$$n_{motor} = M \times CONTAGEM \quad (9)$$

Este valor é armazenado na variável *Vmedida*.

O dispositivo tem que, além de calcular o valor da velocidade do motor, verificar a todo instante se este valor é menor ou maior igual ao valor limite que é dado por:

$$V_{limite} = Vn \times AJUSTEFIXO \quad (10)$$

Sendo a variável *AJUSTEFIXO* já estabelecida no programa como sendo 1,1, ou seja, 110% do valor da velocidade nominal, *overspeed* máximo admitido pelo equipamento proposto. A cada vez que a interrupção do *timer 2* é executada, o valor da velocidade é calculado e é feita a verificação se este valor é maior que a velocidade limite:

$$V_{medida} \geq V_{limite} \quad (11)$$

Se esta inequação, em algum momento, tornar-se verdadeira, uma *flag* intitulada *shutdown* passa a ter valor binário igual a 1, o que habilita um bloco de instruções que serão executadas em uma situação de falha.

Durante o desenvolvimento do projeto, chegou-se a conclusão que seria conveniente utilizar um botão de impulso que tivesse a capacidade de habilitar e desabilitar a função de proteção, ou seja, a inequação exposta anteriormente só será verificada se a proteção estiver ativa. Ao alimentar a placa, a proteção, por *default*, está habilitada, mas caso se deseje desabilitá-la, para que possa ser feito um ajuste dos parâmetros com o motor operando sem que um sinal de *shutdown* seja gerado, por exemplo, basta pressionar o botão com tal função.

Quando é gerado o sinal de *shutdown*, uma das instruções que será executada, devido a *flag shutdown* passar a ter valor lógico verdadeiro, é responsável pelo acionamento do relé. O contato normalmente aberto do relé é então fechado e assim permanece até que um sinal de *reset* seja dado, logo adicionou-se um outro botão de impulso para ter a função de gerar este sinal de *reset*, o qual permitirá que a bobina do relé seja desenergizada.

Eventualmente, um sinaleiro, uma buzina, ou um giroflex pode está conectado em série com o contato normalmente aberto deste relé. Logo, em uma situação de falha, estes equipamentos irão emitir os sinais sonoros e visuais que irão alertar o operador sobre a existência da falha, mesmo o motor já estando parado. Para desativar estes sinais, o operador deve então pressionar o botão com função de *reset*.

Estes dois botões foram então conectados aos pinos que puderam ser configurados como interrupções externas (*INT1* e *INT2*, respectivamente). Sempre que ocorrer a variação do sinal fornecido a este pino, evento que ocorre quando o botão de impulso é pressionado, uma interrupção é gerada. Quando a interrupção *INT1* é executada, a *flag_EN* inverte seu nível lógico (zero ou um) e quando a *INT2* é executada, a *flag shutdown* é forçada a ter nível lógico zero (falso).

O código é composto por seis arquivos os quais serão citados de forma sucinta a seguir, mas estão presentes, de forma integral, no APÊNDICE F:

- Arquivo "main.h" (Código 1) no qual foram feitas as principais configurações de operação do dsPIC como, por exemplo, qual *clock* interno, quais pinos são entradas ou saídas digitais, quais são entradas analógicas, etc. Neste arquivo também foram declaradas as variáveis, seus tipos e seus valores iniciais;
- O arquivo intitulado "TIMERx.h" (Código 2) é uma biblioteca auxiliar na qual foram criadas as funções de *delay* de microssegundos e milissegundos que por vezes foram necessárias para controle de tempo;
- É no arquivo "config_interrupts.h" (Código 3) onde são encontrados todas as atribuições aos bits dos registradores que configuram as interrupções utilizadas (interrupções externas, interrupção dos *timers* e do conversor analógico-digital). Diante da grande quantidade de registradores, evitou-se fazer um detalhamento minucioso neste texto, mas comentários foram inseridos em todas as linhas do código. Todas as atribuições foram baseadas em MICROCHIP (2005);
- O bloco de comando que cada interrupção irá executar quando for chamada está apresentado no arquivo "INTERRUPTS.h" (Código 4). Cada interrupção possui sua *flag* específica. A *flag* do *timer* 2, por exemplo, é intitulada *T2IF* e esta passa a ter valor unitário (verdadeiro) quando o *timer* 2 estoura (*TMR2* tem valor igual a *PR2*). Neste momento, o bloco da função *_ISR _T2Interrupt ()* é executada e, como se sabe, é justamente neste bloco que, a cada 500 ms, são calculados os valores da velocidade do motor (Equação 9), verificada as condições de falha (Equação 11), dentre outras instruções;
- A biblioteca "LCD_C30.h" (Código 5) é responsável por conter as funções que são utilizadas para escrever no LCD como, por exemplo, *LIMPA_LCD()* que apaga tudo que estiver escrito, *posicao_lcd(x,y)* que posiciona o cursor na linha *x* e coluna *y* para que o próximo caractere a ser escrito seja alocado nesta posição, *FRASE ("rpm")* que escreve no *display* justamente o texto entre aspas duplas e *EN_LCD (int n, char ndig, char mode, char punkt)* que escreve o número inteiro contido na variável *n*, o qual deve ser composto de *ndig* dígitos. Deve ser informado para a função, através da variável *mode*, se deve ser inserido sinal (+ ou -) ao expor número e qual o sinal. Se o número for decimal, deve ser informada também a posição da vírgula através da variável *punkt*;
- O arquivo "main.c" (Código 6) é o principal arquivo do código pois é nele onde se define qual o dsPIC utilizado, onde são feitos os links com as bibliotecas citadas anteriormente, onde está a função *main ()* e é dentro desta função que existe a linha de código *while (1)*

responsável pelo *loop* infinito de execução do código. As funções de escrita do *display* estão definidas dentro deste *loop* de forma a evitar qualquer atraso em alguma rotina de interrupção. No *display* são impressos, na primeira linha, os valores da velocidade nominal do motor, V_n , e do múltiplo M , ambos definido pelo usuário. Na segunda linha, são impressos o valor da velocidade do motor calculada a cada 500 ms, que está armazenado na variável V_{medida} , e, ao final da segunda linha, o *status* da proteção, se está habilitada ou não, que pode ser verificado através do textos: *PROT:ON* ou *PROT:OFF*.

6.1.3 Ensaios com gerador de frequência

Para realizar os ensaios iniciais, foi desenvolvido um gerador de frequência com o dsPIC30F2010 o qual foi programado para gerar uma forma de onda com a frequência definida por um teclado numérico e mostrar este valor no *display* LCD. O valor da frequência é digitado (são permitidos valores de 200 a 6000 Hz) e deve ser pressionada a tecla com função de *enter*, conforme pode ser verificado na Figura 30. O sinal de frequência, disponibilizado no pino de *pwm* do dsPIC pode então ser inserido na placa do projeto nos bornes *RPM+* e *RPM-*.

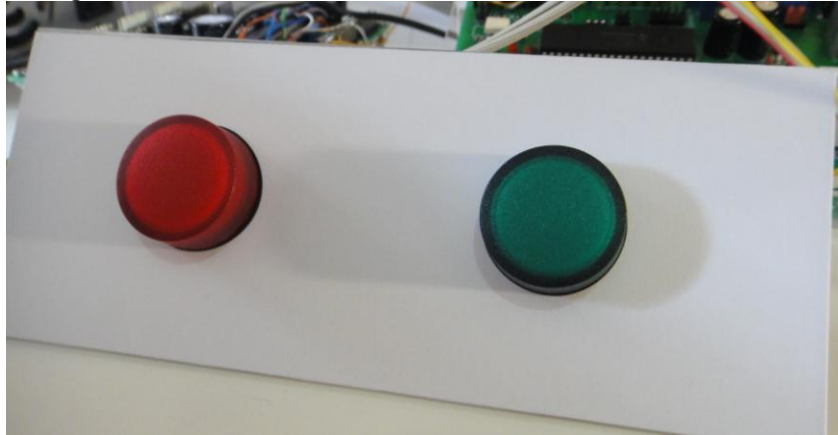
Figura 30 - Gerador de frequência desenvolvido para testes.



Fonte: Produção do próprio autor.

A fim de fazer o teste do relé, utilizaram-se dois sinaleiros, um verde e um vermelho, ambos de 12 V contínuos, que foram inseridos em série, respectivamente, com os contatos NF e NA, conforme pode ser observado na Figura 31. O esquemático das ligações feitas entre da placa e os acessórios citados pode ser averiguado na Figura 48, APÊNDICE C.

Figura 31 - Painel com sinaleiros de 12 VCC/VCA do fabricante ABB.



Fonte: Produção do próprio autor.

Inicialmente foi simulada a operação de um motor com velocidade nominal de 1800 rpm e cremalheira de 152 dentes. A velocidade nominal foi configurada através do potenciômetro 1 e o múltiplo foi calculado, conforme a equação abaixo para, logo em seguida, ser configurado através do potenciômetro 2:

$$M = \frac{120}{n_{teeth}} = \frac{120}{152} = 0,79 \quad (12)$$

Feitas as configurações necessárias. Podem ser feitas as simulações para diferentes velocidades de operação do motor. Estando operando com 1800 rpm, um sensor de velocidade instalado na cremalheira geraria uma forma de onda com a frequência de:

$$f_{pickup} = \frac{n_{motor}}{60} \times n_{teeth} = \frac{1800rpm}{60} \times 152 = 4560 \text{ Hz} \quad (13)$$

Foi definida esta frequência de operação no gerador de frequência e verificou-se a correta leitura de velocidade, conforme pode ser comprovado na Figura 49, APÊNDICE C.

Outros testes foram realizados para velocidades diferentes do motor. Para tal, novamente calcularam-se as frequências de simulação para as velocidades de 500 rpm e 1980 rpm, respectivamente:

$$f_{pickup} = \frac{n_{motor}}{60} \times n_{teeth} = \frac{500 \text{ rpm}}{60} \times 152 = 1266 \text{ Hz} \quad (14)$$

$$f_{pickup} = \frac{n_{motor}}{60} \times n_{teeth} = \frac{1980 \text{ rpm}}{60} \times 152 = 4560 \text{ Hz} \quad (15)$$

Nas Figura 50 e Figura 51 (APÊNDICE C) podem ser interpretados os resultados das simulações. O segundo caso, para o qual foi utilizado o valor de 1980 rpm (valor 10% superior a velocidade nominal de 1800 rpm), representa uma situação de falha que pode ser observada através dos seguintes acontecimentos: *display* mostrando intermitentemente letreiro "!!!!OVERSPEED!!!!"; LED2 aceso; sinaleiro vermelho, que está ligado em série com contato NA do relé, aceso; e sinaleiro verde, que está ligado em série com contato NF do relé, desligado.

Diante dos resultados, conclui-se que o código proposto é adequado e o princípio de funcionamento geral está operando conforme proposto.

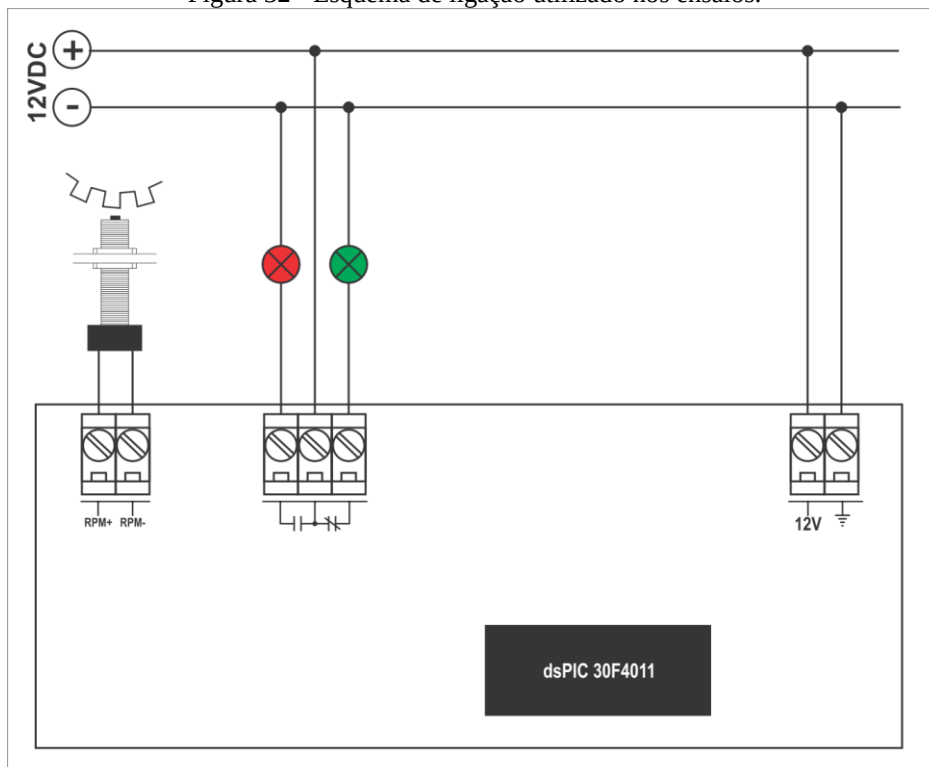
6.1.4 Ensaios com *pickup* magnético

Para verificar qual o comportamento do circuito utilizando um *pickup* magnético, haja vista a mudança das formas de onda e alteração dos valores de pico, uma nova bancada foi configurada sendo esta composta de um motor com velocidade nominal de 1800 rpm; de uma cremalheira de 60 dentes a qual foi instalada no eixo do motor; do sensor magnético; e da placa desenvolvida para o projeto.

As ligações executadas podem ser compreendidas através da Figura 32. Na Figura 33, pode ser observada uma imagem da bancada utilizada para os ensaios. Como pode ser visto, não foi feito o encaixe do *pickup*, este foi posicionado manualmente somente nos momentos de teste.

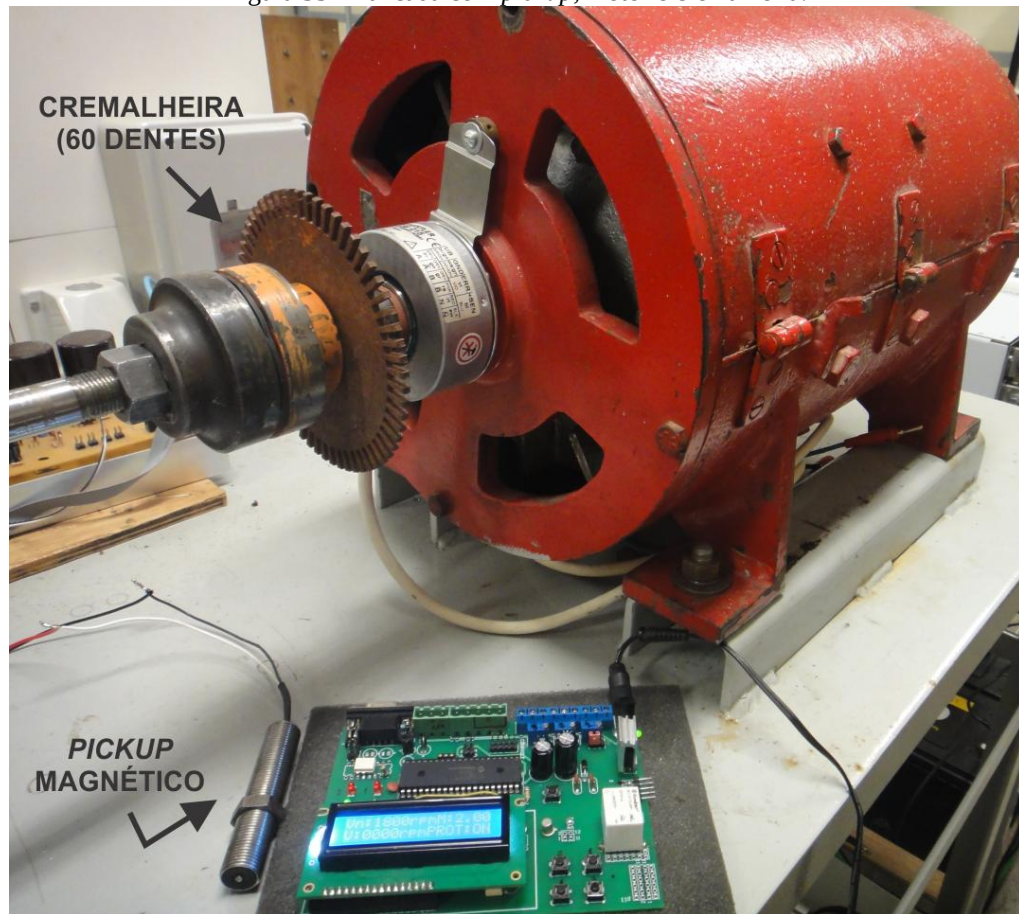
Para que pudesse ser feita uma análise, com o osciloscópio, foi coletada a forma de onda fornecida pelo *pickup* magnético sem que este estivesse ligado a placa. Verificou-se uma mudança dos valores de pico para diferentes velocidades e principalmente para diferentes distancias entre o sensor e a cremalheira, como já havia sido comentado anteriormente.

Figura 32 - Esquema de ligação utilizado nos ensaios.



Fonte: Produção do próprio autor.

Figura 33 - Bancada com *pickup*, motor e cremalheira.



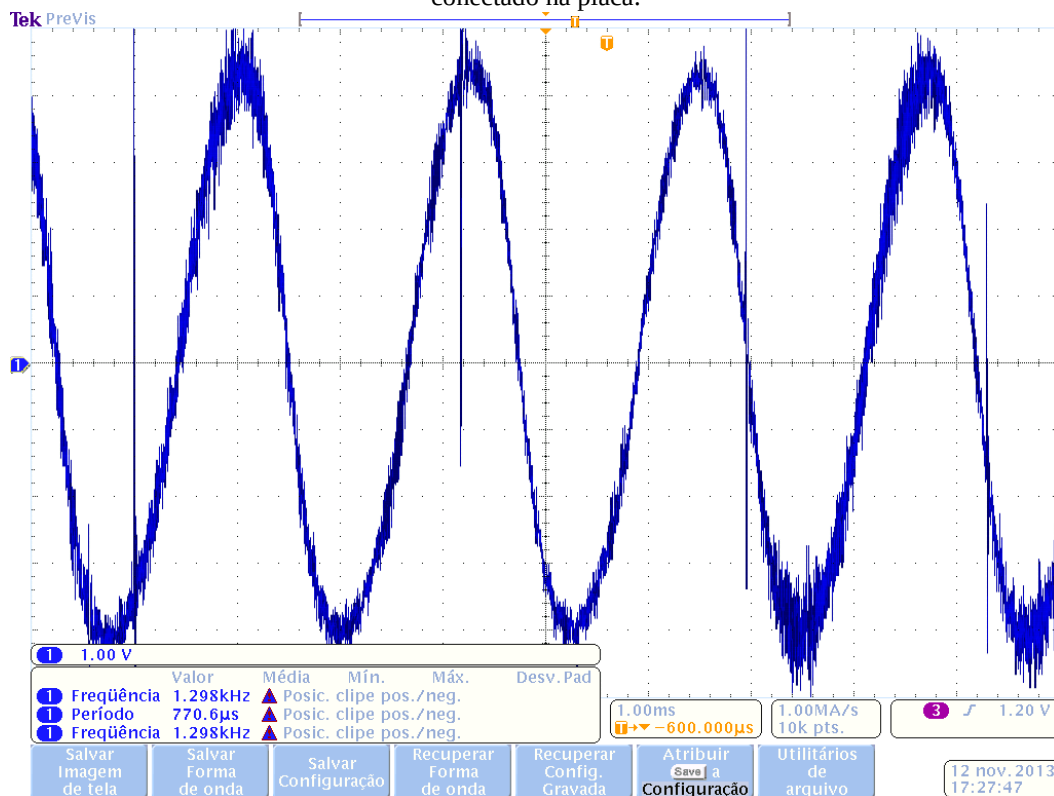
Fonte: Produção do próprio autor.

O motor foi acionado em baixa rotação devido a limitações da própria bancada e esta rotação foi mantida fixa para a realização do ensaio. A Figura 34 representa o sinal coletado neste *pickup* magnético, o qual apresentou aproximadamente 8 V pico-a-pico.

Para compreender melhor as características do sensor utilizado, a resistência do sensor foi medida e apresentou valor de 1125 Ohms o que acaba garantindo maior proteção ao optoacoplador, visto que a corrente será bem reduzida. Este fato verificado dispensaria até o uso da resistência inserida na placa, o que não foi feito devido ao fato de poderem ser feitos testes com outros sensores que possam apresentar baixa impedância e até mesmo geradores de função como foi mostrado anteriormente.

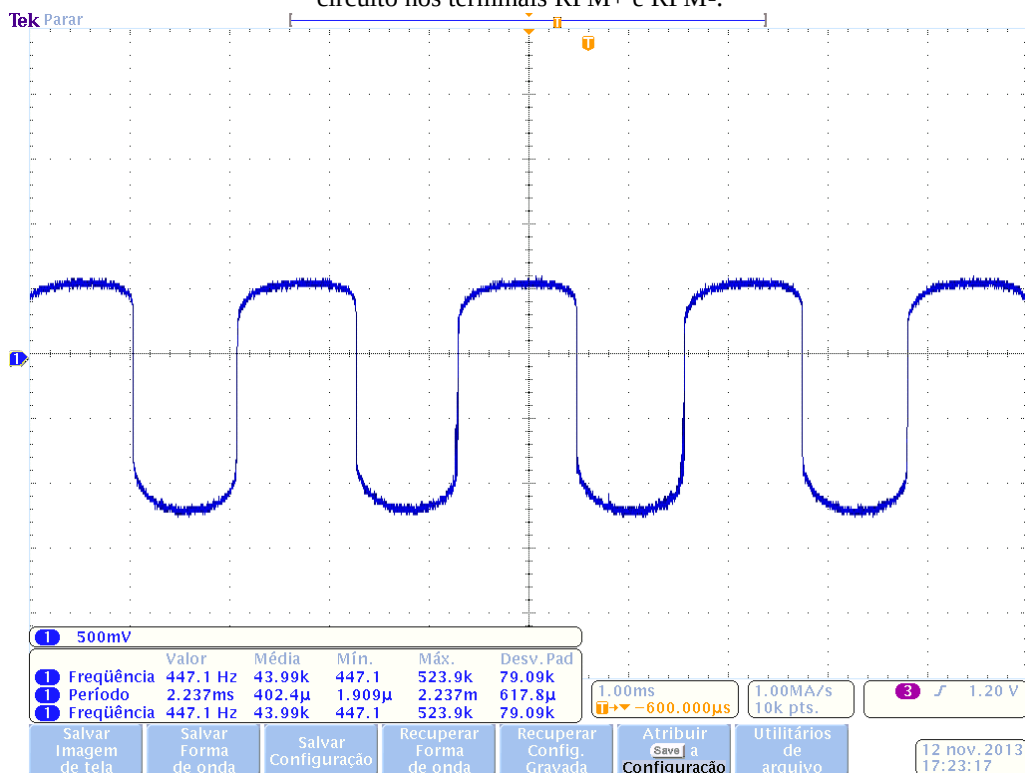
O sinal do sensor, depois de estudado, pôde ser conectado, através dos terminais RPM+ e RPM-, a placa. Com a referência do osciloscópio conectada ao terminal RPM+ e a ponta de prova ao RPM-, foi feita uma coleta do sinal o qual pode ser visto na Figura 35. O sinal apresenta valor positivo de 0,7 V quando o diodo da entrada está conduzindo (semi-ciclo negativo do sinal gerado pelo *pickup*) e valor negativo de aproximadamente 1,4 V quando o LED do optoacoplador está sendo acionado (semi-ciclo positivo).

Figura 34 - Sinal de onda coletado pelo osciloscópio com ponteira nos terminais do *pickup* sem que ele esteja conectado na placa.



Fonte: Produção do próprio autor.

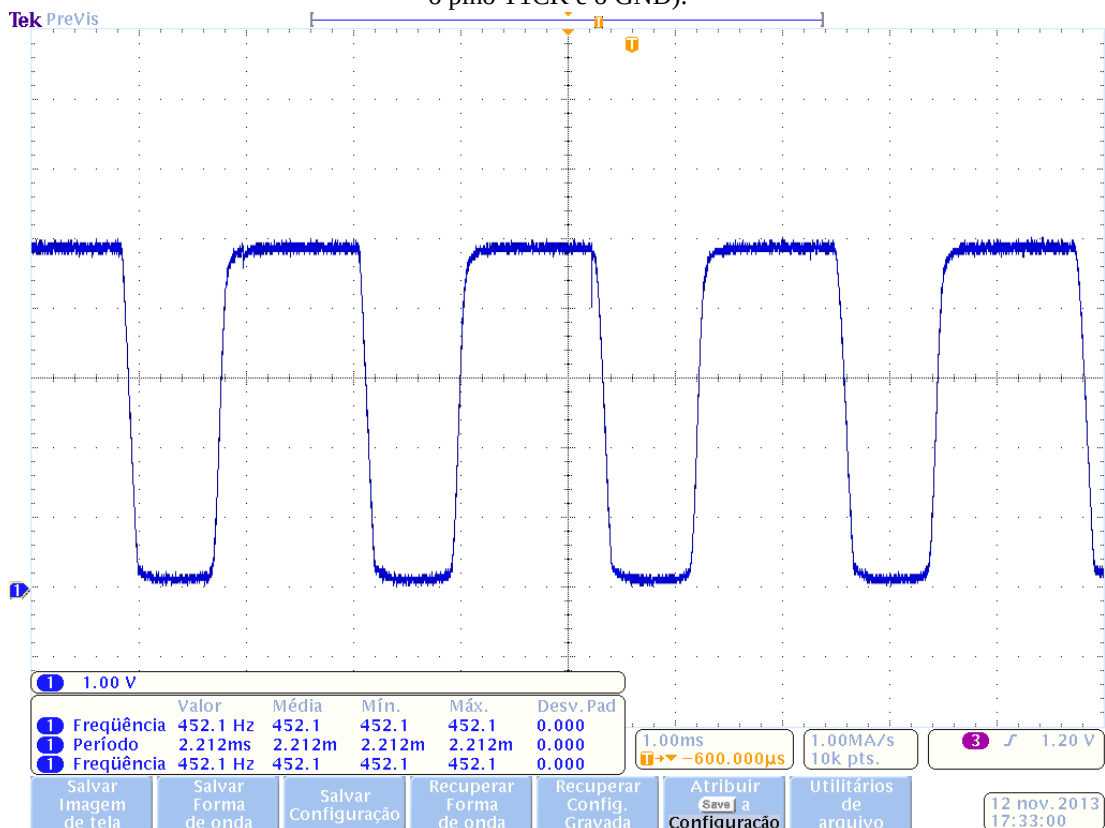
Figura 35 - Sinal de onda coletado pelo osciloscópio com ponteira nos terminais do *pickup* com este inserido no circuito nos terminais RPM+ e RPM-.



Fonte: Produção do próprio autor.

Outro sinal foi coletado, sendo este agora na saída do optoacoplador (Figura 36), o qual, como era previsto, apresentou valores nulos que coincidem com o momento no qual o LED do optoacoplador está acionado e polariza o transistor, levando o sinal de GND para o pino T1CK e também apresentou valores iguais a 5 V que ocorrem na situação contrária, quando o transistor não está polarizado.

Figura 36 - Sinal de onda coletado pelo osciloscópio com ponteira nos terminais de saída do optoacoplador (ente o pino T1CK e o GND).



Fonte: Produção do próprio autor.

Para que o protótipo possa fornecer o valor da velocidade, o valor do múltiplo tem que ser ajustado, sendo a cremalheira de 60 dentes, através do potenciômetro com o valor de:

$$M = \frac{120}{n_{teeth}} = \frac{120}{60} = 2 \quad (16)$$

O valor do múltiplo foi configurado e, para que seja verificado se o valor mostrado no *display* do protótipo está correto, pode ser calculado o valor da velocidade visto que se tem a frequência do sinal gerado pelo *pickup* (aproximadamente 446 Hz, conforme pode ser visto nas formas de onda coletadas pelo osciloscópio):

$$n_{motor} = \frac{60 \times f_{pickup}}{n_{teeth}} = \frac{60 \times 446}{60} = 446 \text{ rpm} \quad (17)$$

Na Figura 37, pode ser vista uma imagem do *display* no qual é exposto o valor da velocidade calculada pelo controlador. O valor é igual ao teórico validando novamente o código programado no dsPIC.

Figura 37 - Leitura feita pelo protótipo paralelamente as coletas do osciloscópio.



Fonte: Produção do próprio autor.

6.2 Implementação do segundo protótipo

Desenvolvido o código e realizado os testes os quais foram detalhados nas seções anteriores, pôde ser definido o que realmente foi necessário para desenvolver o protótipo proposto, logo se iniciou uma nova etapa de desenvolvimento a fim de ser produzido um equipamento com o menor tamanho e custo possível.

Iniciaram-se as mudanças com a troca do dsPIC. Não houve mais a necessidade do uso de um dsPIC de 40 pinos visto que são efetivamente utilizados 12 pinos (6 para o *display*, 2 para os potenciômetros, 2 para os *push-buttons*, 1 para acionar o relé e um para ler o sinal do sensor de velocidade).

Um novo esquemático foi desenvolvido utilizando agora o dsPIC33FJ12MC202 o qual possui 28 pinos, sendo 21 deles configuráveis como entrada ou saída. Com o auxílio do software *Altium Designer*, um novo esquemático foi desenvolvido (Figura 52, APÊNDICE D), no qual pode ser visto as ligações entre periféricos necessários para o novo protótipo e o dsPIC.

Como pode ser visto, apenas um LED foi utilizado o qual sinaliza o funcionamento da fonte de alimentação do dsPIC. Todas as informações podem ser verificadas no *display* LCD que foi configurado exatamente da mesma forma como no protótipo anterior, ou seja, na primeira linha são dispostos os valores da velocidade nominal e do múltiplo, ambos configuráveis pelo usuário, e, na segunda linha, o valor da velocidade medida a cada instante assim como o *status* da proteção (se está habilitada ou não). Em uma situação de falha, o *display* deixa de expor estes valores e pisca na tela o texto "!!!!OVERSPEED!!!!".

Continua sendo através dos dois potenciômetros que o usuário define os valores da velocidade nominal e do múltiplo. Também foi mantida a função dos dois botões, um para habilitar e desabilitar a proteção de *overspeed* e o outro para reconhecer a falha fazendo com que o relé seja desenergizado.

6.2.1 A placa de circuito impresso

O layout da placa de circuito impresso foi desenvolvido também no *Altium Designer* (Figura 53, APÊNDICE D). É neste layout onde podem ser vistas as ligações

elétricas que devem ser feitas na placa a ser confeccionada de modo a se ter um circuito com a finalidade proposta.

Através de uma integração entre os softwares *Altium Designer* e o *SolidWorks*, ambos capazes de trabalhar com projeções 3D, foram geradas visualizações nas quais podem ser vistas a disposição real e a dimensão relativa dos componentes. A Figura 54 e a Figura 55, ambas presentes no APÊNDICE D, contemplam uma visualização da placa de ambos os lados.

Com relação aos componentes, uma diferença existe entre o esquemático desenvolvido para o dsPIC30F4011 e o dsPIC33FJ12MC202 visto que este último deve ser alimentado com uma tensão contínua entre 3 e 3,6 V (MICROCHIP, 2007). Logo, uma placa de aproximadamente 15x10 mm foi confeccionada com o regulador de tensão de 3,3 V, LP3982 SMD da National Semiconductor. A placa desenvolvida pode ser vista na Figura 56 do APÊNDICE D e deve ser inserida na placa principal através do borne *REG3V3*.

A Tabela 5 (APÊNDICE D) exhibe a lista dos materiais necessários à elaboração da placa. A primeira coluna enumera os nomes dos componentes apresentados no esquemático preparado no *Altium Designer*; a segunda coluna se trata da especificação técnica destes componentes; e as demais a quantidade e a imagem de cada um.

6.2.2 Códigos

Os mesmos periféricos internos do microcontrolador configurados para o dsPIC30F4011 foram configurados no dsPIC33FJ12MC202: o *timer* 1 opera como base de tempo de 500 ms; o *timer* 2 é um contador de eventos; o *timer* 3 é base de tempo das funções de *delay*, a interrupção externa 1 é chamada ao ser pressionado o botão que habilita e desabilita a proteção; a interrupção externa 2 força a *flag shutdown* a ter valor lógico falso; e o conversor analógico-digital configurado para ler duas entradas analógicas advindas dos potenciômetros que irão definir o valor da velocidade nominal e do múltiplo, que está diretamente relacionado com o número de dentes da cremalheira.

Alguns registradores deste dsPIC não possuem a mesma nomenclatura do dsPIC anterior, portanto, foram adequados os códigos e estes (Código 7 ao Código 12) podem ser conferidos no APÊNDICE G.

Este dsPIC utiliza 2 pulsos do *clock* principal do sistema para gerar 1 ciclo de máquina e a fonte de *clock* utilizada continuou sendo a de 7,37 MHz, fornecida pelo cristal

interno do dsPIC, mas não foi configurado nenhum multiplicador. Logo, a velocidade de processamento configurada é de:

$$F_{cy} = \frac{F_{osc} \times PLL}{2} = \frac{7,37 \text{ MHz} \times 1}{2} = 3,69 \text{ MHz} = 3,69 \text{ MIPS} \quad (18)$$

E o ciclo de máquina (CM), ou período T_{cy} do *clock* da máquina, passou a ser de:

$$CM = T_{cy} = \frac{1}{F_{cy}} = \frac{1}{3,69 \text{ MHz}} = 271,37 \text{ ns} \quad (19)$$

Como o *timer* 1 deve executar um bloco de comandos a cada $T_{timer1} = 500 \text{ ms}$, o valor do registrador *PR1* tem que ser igual a:

$$PR1 = \frac{T_{timer1}}{T_{cy} \times prescale} = \frac{500 \text{ ms}}{271,37 \text{ ns} \times 256} = 7196 \quad (20)$$

6.2.3 Simulações

Como a placa desenvolvida para o equipamento não foi confeccionada, apenas projetada, uma simulação foi feita com o auxílio da ferramenta *ISIS* do software *Proteus* na qual é possível inserir um componente programável assim como seu código. O esquemático utilizado neste software (Figura 57, APÊNDICE D) é praticamente igual ao esquemático apresentado no *Altium Designer*. O arquivo com extensão *.hex* ou *.cof* gerado pelo *MPLAB* foi carregado no dsPIC e a simulação foi feita utilizando o bloco gerador senoidal do *Proteus*.

O gerador foi ajustado para gerar uma frequência de 4560 Hz e o múltiplo ajustado para 0,79 afim de ser feita uma simulação de uma cremalheira de 152 dentes. Este mesmo ensaio foi realizado na placa do dsPIC30F4011 e os valores da frequência de 4560 Hz e do múltiplo de 0,79 justificados, por isso não serão repetidos. No resultado da simulação realizada no PROTEUS (Figura 58, APÊNDICE D), pode ser verificado que o valor da velocidade foi calculado corretamente e não está com valor 110% acima da velocidade nominal, logo o sinal de *shutdown* não foi gerado e o relé encontra-se desenergizado.

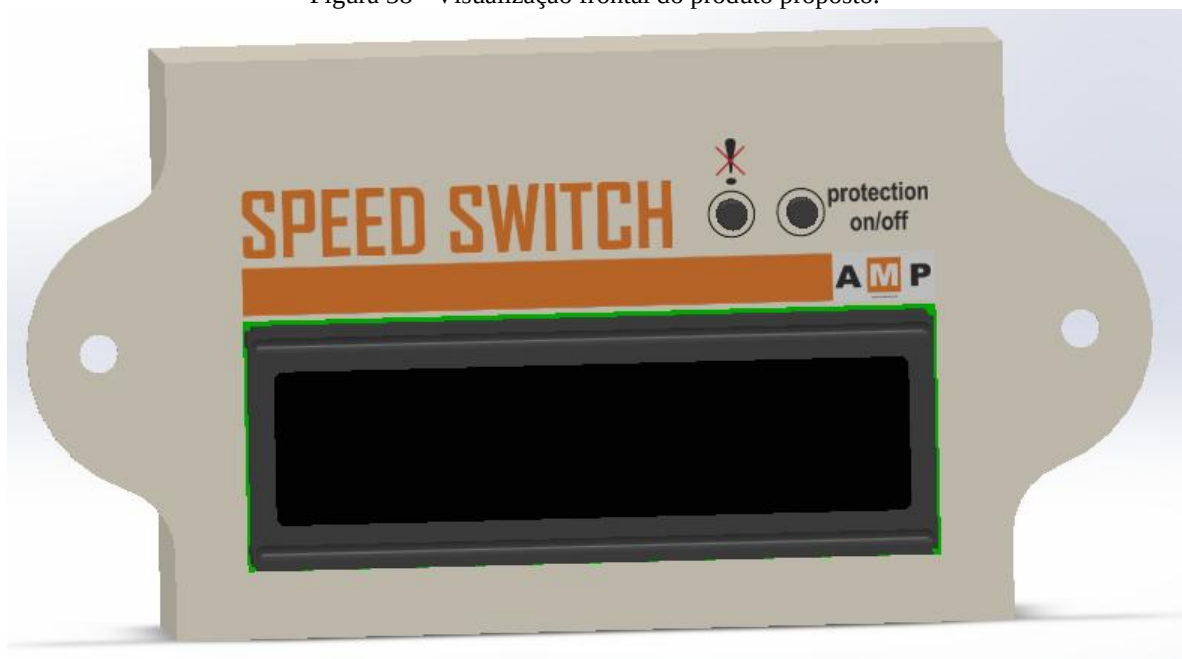
Já a Figura 59 (APÊNDICE D) apresenta o resultado de uma simulação na qual o valor da frequência do gerador utilizado foi mantida no valor de 4560 Hz, mas o valor nominal da velocidade foi sendo reduzido até atingir um valor menor que 1636 rpm, ou seja, o valor da velocidade calculada (1800 rpm) passou a ser maior que 110% do valor de 1636 rpm e o sinal de *shutdown* foi gerado. Isto pode ser confirmado observando-se que o *display* deixou de expor os valores das variáveis e começou a piscar o texto "!!!!OVERSPEED!!!!" e além disto, pode ser visto que o relé foi energizado.

6.2.4 Produto final

Com o intuito de apresentar o projeto na forma de um produto que possa vir a ser confeccionado e instalado em circuitos de proteção do motor, foi desenvolvido, com o auxílio do software *SolidWorks*, um invólucro no interior do qual está a placa de circuito impresso. O conjunto foi desenvolvido de forma que o usuário só possa ter acesso ao borne de alimentação, ao borne que acessa o contato 1NAF do relé, ao borne de conexão do sensor de velocidade e aos potenciômetros de ajuste.

A Figura 38 apresenta uma visualização frontal do produto proposto. Muitas outras imagens foram produzidas e serão referenciadas a seguir, mas estão presentes no APÊNDICE D.

Figura 38 - Visualização frontal do produto proposto.



Fonte: Produção do próprio autor.

Como pode ser visto nas Figura 60 e Figura 61, que apresentam a vista frontal e posterior, uma atenção foi dada a criação de etiquetas de identificação que facilitem a identificação dos componentes com os quais o operador terá que trabalhar.

As Figura 62 e Figura 63 contemplam o produto de dois ângulos distintos e a Figura 64 apresenta uma visão ampliada contemplando os bornes de conexão e os potenciômetros.

As características deste segundo protótipos foram concentradas de forma resumida no panfleto apresentado pela Figura 65. Nele podem ser vistas tanto características dimensionais, como características elétricas.

6.2.5 Exemplo de aplicação

O equipamento desenvolvido não deixa de ser um sensor de velocidade que, juntamente com os outros sensores já apresentados (pressão e temperatura), podem compor um circuito que confira todas as funções de proteção exigidas para o motor.

Um circuito de comando foi concebido com os sensores determinando o funcionamento do solenóide de combustível o qual libera a passagem de combustível quando é energizado (ver Figura 66, APÊNDICE E). Uma chave denominada *ON/OFF* é utilizada cujo propósito é justamente a liberação de combustível antes da *CHAVE DE IGNIÇÃO* ser pressionada.

Em condições normais de operação (pressão do óleo, temperatura e velocidade dentro dos limites), todos os sensores possuem seus contatos abertos não permitindo que a bobina *K1* seja energizada e, assim, o solenóide continua energizado (liberando combustível). Ao ocorrer uma falha, o contato do respectivo sensor que a detectou fecha e a bobina *K1* é energizada. Assim, o contato desta bobina que era normalmente fechado agora abre; deixa de haver tensão no solenóide de combustível; e não é mais fornecido combustível para o motor forçando a parada.

Um circuito de alarme, composto por uma buzina, um relé auxiliar com um contato reversível 1NAF e um botão de *reset*, foi inserido no circuito. A buzina é acionada juntamente com a bobina *K1*. A bobina *K1* permanece energizada após ter ocorrido uma falha. Isto ocorre porque, mesmo que a falha tenha sido por uma elevação de temperatura, ou seja, o sensor volte a estar normalmente aberto quando a temperatura diminuir, o motor não estará mais operando após a falha e o sensor de pressão vai possuir o contato fechado, permitindo a

continuidade do fornecimento de tensão à bobina *K1*. Mas o operador deve ter o direito de desligar o circuito de alarme mesmo sem ter que desligar todo o painel elétrico, logo o botão *RESET* foi inserido de forma que, ao ser pressionado, um caminho alternativo para a corrente passe a existir e o sinal sonoro seja desativado (observar Figura 66, APÊNDICE E).

No momento da partida, não existe pressão no óleo de lubrificação, ou seja, o sensor de pressão possui seus contatos fechados. Desta forma, a bobina *K1* estaria sempre energizada e o motor não conseguiria partir devido à ausência de combustível. Para contornar este problema, um relé temporizador, *RT*, foi inserido logo após a chave *ON/OFF* e seu contato, normalmente aberto, alocado em série com o sensor de pressão. Assim que o circuito é energizado e a partida do motor vai ser efetuada, o contato do relé temporizador está aberto, logo a bobina *K1* não possuirá tensão. Após algum tempo, suficiente para que o motor já tenha partido, o contato do relé de tempo fecha habilitando a função de proteção por perda de pressão. A partir de então, qualquer falha, acarretará a parada do motor.

O dispositivo de *overspeed* possui um relé na saída com contato 1NAF mas, para este exemplo de aplicação, o contato normalmente aberto deve ser utilizado o qual fechará sempre que a velocidade do motor ultrapassar o limite de 110% do seu valor nominal.

7 CONCLUSÃO

Muitas das instalações elétricas de embarcações antigas não mais atendem às exigências normativas de segurança em vigor. Logo, surgiu no mercado uma demanda potencial de serviços para adaptação tecnológica (*retrofit*) destas instalações.

O *retrofit* nada mais é do que a substituição de equipamentos ultrapassados por tecnologias de ponta. Sistemas embarcados de controle e monitoramento têm sido, então, utilizados para diversos tipos de aplicações. O controlador da ComAp, apresentado neste trabalho, que tem função de garantir todas as proteções mecânicas do motor, é um exemplo disto. O equipamento proposto neste trabalho também é um modelo de sistema embarcado, o qual foi desenvolvido com objetivo de efetuar somente proteção contra *overspeed* e, para tal, foi utilizado um microcontrolador dsPIC.

Simulações e ensaios foram realizados até se chegar a conclusão de que o equipamento cumpre o seu objetivo de proteção; é confiável; sensível; seguro; e versátil, visto que podem ser conectados sensores de velocidade do tipo *pickup* magnético, digitais e até *encoders*.

No final do trabalho, foi apresentado um exemplo de aplicação no qual foram utilizados relés, botões, sinaleiros e os sensores de pressão, temperatura e velocidade (composto pelo sensor de velocidade e pelo dispositivo de proteção proposto). Apesar de se tratar de uma aplicação de baixo custo, esta se mostra confiável e capaz de atender a demanda de mercado que incentivou o seu desenvolvimento.

Por fim, espera-se que o equipamento proposto possa, em breve, ser comercializado e considerado uma contribuição relevante, fazendo jus ao esforço aplicado em seu desenvolvimento. E que cada vez mais possam ser feitas modificações de modo a torná-lo mais adequado às necessidades. A seguir, serão mencionadas algumas sugestões, da própria autora, para melhorias que não foram implementadas, algumas devido a limitações temporais, outras simplesmente para evitar desvios na linha da pesquisa e desenvolvimento.

7.1 Sugestões para trabalhos futuros

O equipamento apresentado foi construído de forma a apresentar o menor custo e simplicidade possível. Para tal, foram utilizados potenciômetros através dos quais podem ser determinados os valores da velocidade nominal e do múltiplo (constante que depende da

quantidade de dentes da cremalheira acoplada no volante) do motor. Em vez de serem utilizados potenciômetros, poderia ser inserida uma quantidade maior de botões (*push-buttons*) com funções variadas como, por exemplo, para incremento, decremento, *enter*, etc. Isto deixaria o sistema completamente digital.

Estes ajustes que têm de ser feitos pelo usuário e no próprio equipamento também poderiam ser feitos através de uma comunicação sem fio, via *bluetooth*, entre o equipamento e um *smartphone* que opere com plataforma *Android*. Ao equipamento só teria que ser adicionado um módulo *bluetooth* que é conectado ao microcontrolador através dos pinos *RX* e *TX* e se comporta como um dispositivo serial padrão. A empresa *Shenzhen Jiayuan Electronic Co. Ltd.* fabrica o *bluetooth* modelo JY-BT03 que são adquiridos no Brasil a custo praticamente irrelevante. Logo, esta melhoria funcional resumir-se-ia ao desenvolvimento de um aplicativo para o *Android* no qual os dados são digitados e enviados via *bluetooth*.

O dispositivo foi desenvolvido com o objetivo de efetuar proteção garantindo que nunca ocorra uma situação na qual a velocidade do motor ultrapasse valores maiores ou iguais a 110% do valor nominal do motor. Quanto à proteção de velocidade, algumas sugestões podem ser dadas. Este valor percentual poderia passar a ser mais um ajuste a ser feito pelo operador. Poderia ser adicionado um modo de alarme para o qual um relé é energizado quando a velocidade atinge 105% do valor nominal, por exemplo. Bastaria ser adicionado um novo relé a placa e mais uma linha de código que verificaria esta condição. Verificações de *underspeed* também seriam interessantes de serem verificadas. Em linhas de código, bastaria inverter a inequação para que o sinal de *shutdown* fosse gerado quando a velocidade medida fosse menor ou igual a um valor determinado pelo usuário. Resumindo, poderiam existir quatro variáveis com valores percentuais ajustáveis pelo usuário: *%underspeed_alarm*, *%underspeed_shutdown*, *%overspeed_alarm* e *%overspeed_shutdown*. Em termos de *hardware*, quatro relés deveriam ser adicionados à placa.

Conclui-se que nada impede que este protótipo seja aperfeiçoado e possa evoluir de forma a se tornar um controlador apto a garantir todas as proteções mecânicas exigidas, ou seja, controlar-se-ia não só velocidade, mas também pressão e temperatura.

REFERÊNCIAS BIBLIOGRÁFICAS

AMERICAN BUREAU OF SHIPPING. *Guide for automatic or remote control and monitoring for machinery and systems (other propulsion) on offshore installations*. Houston, 2003.

ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE CONSTRUÇÃO NAVAL E OFFSHORE. **Dados do Setor**. Disponível em: <<http://www.abenav.org.br>> Acesso em: 02 set. 2013.

BOSCH. **Sistemas elétricos e eletrônicos para veículos automotivos - Alternadores**. Disponível em: <<http://www.joseclaudio.eng.br/geradores/PDF/diesel1.pdf><http://ava.mecatronica.org.br/claroline/claroline/backends/download.php?url=L0NhcmdhXyZfUGFydGlkYS9BbHRlcm5hZG9yZXMucGRm&cidReset=true&cidReq=INS323>>. Acesso em: 05 out. 2013.

BRITO, N; RIBEIRO, L; ESTEVES, J. S. *Electric Power Generating Bicycle*. In: *International Conference on Hands-on Science*, 3, 2006, Braga. **Science Fairs...** Braga: University of Minho, 2006. Disponível em: <<http://repositorium.sdum.uminho.pt/bitstream/1822/6056/1/158.pdf>>. Acesso em: 05 out. 2013.

BUREAU VERITAS. *Rules for the classification of steel ships - Machinery, Electricity, Automation and Fire Protection*. Paris, 2007.

ComAp. **InteliDrive Lite - Operator Guide**. Prague, 2012. Disponível em: <http://cdn.comap.cz/files_tmp/manuals/ID-FLX%20Lite-1.3-OperatorGuide.pdf<http://www.joseclaudio.eng.br/geradores/PDF/diesel1.pdf>>. Acesso em: 21 out. 2013.

ComAp. **InteliDrive Lite - Reference Guide**. Praha, 2013. Disponível em: <http://cdn.comap.cz/files_tmp/manuals/ID-FLX-Lite-1.7%20Reference%20guide.pdf<http://www.joseclaudio.eng.br/geradores/PDF/diesel1.pdf>>. Acesso em: 21 out. 2013.

CUMMINS. **Grupos Geradores acionados por Motor Diesel. C110D6**. Guarulhos, 2011. Disponível em: <http://www.cumminspower.com.br/pdf/spec-sheets/6B/PT/60Hz/C110D6_PT_Rev08.pdf>. Acesso em: 23 out. 2013.

EMÍLIO, J. **Regulamentos e Direito Marítimo - Classificação de Navios**. Paço d'Arcos: Escola Superior Náutica Infante D. Henrique, [2009]. Disponível em: <<http://www.enautica.pt/publico/professores/jemilio/pdf/REG&DIR/Classifica%C3%A7%C3%A3o%20de%20navios-V2.pdf>> Acesso em: 03 set. 2013.

FILHO, G. D. V. **Avaliação das instalações de máquinas em navios visando redução do uso de combustível fóssil**. 2011. 103 f. Dissertação (Mestrado em Engenharia) – Departamento de Engenharia Naval e Oceânica, Escola Politécnica da Universidade de São Paulo, São Paulo, 2011.

FITZGERALD, A. E.; JUNIOR, C. K.; UMANS, S. D. **Máquinas Elétricas - Com introdução à eletrônica de potência**. 6 ed. Porto Alegre: Bookman, 2006.

FONSECA, M. M. **Arte Naval. Volume I**. 6. ed. Rio de Janeiro: Serviço de Documentação da Marinha, 2002.

GANESAN, V. **Internal combustion engines**. 3. ed. New Delhi: Tata McGraw-Hill, 2007.

GRUPO AYRESTECH. **Sensor Magnético PK3103**. 2013. Disponível em: <http://www.ayrestech.com.br/site_br/produtos.php>. Acesso em: 6 out. 2013.

GUPTA, H. N. **Fundamentals of internal combustion engines**. 2. ed. New Delhi: Prentice Hall of India, 2013.

INFOMOTOR. **Componentes do sistema de arrefecimento**. 2009a. Disponível em: <<http://www.infomotor.com.br/site/2009/05/componentes-do-sistema-de-arrefecimento/>>. Acesso em: 13 set. 2013.

INFOMOTOR. **Componentes do sistema de lubrificação**. 2009b. Disponível em: <<http://www.infomotor.com.br/site/2009/03/componentes-do-sistema-de-lubrificacao/>>. Acesso em: 13 set. 2013.

INFOMOTOR. **Funcionamento do sistema de lubrificação**. 2009c. Disponível em: <<http://www.infomotor.com.br/site/2009/03/funcionamento-do-sistema-de-lubrificacao/>>. Acesso em: 13 set. 2013.

INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **IEEE Std 45: Recommended Practice for Electrical Installations on Shipboard**. New York, 2002.

JESUS, C. G.; GITAHY, L.M. Transformações na Indústria de Construção Naval Brasileira e seus Impactos no Mercado de Trabalho (1997-2007). In CONGRESSO DE DESENVOLVIMENTO REGIONAL DE CABO VERDE, 1., 2009, Cabo Verde. **Cabo Verde: Redes e Desenvolvimento Regional**. Disponível em:

<<http://www.apdr.pt/congresso/2009/pdf/Sess%C3%A3o%2039/79A.pdf>>. Acesso em: 02 set. 2013.

MARINHA DO BRASIL. Diretoria de Portos e Costas. **Máquinas e sistemas de propulsão PRO 01**. 01. ed. Rio de Janeiro: Unidade de Estudo Autônomo, 2008.

MICROCHIP. **DS51795A: PICkit 3 Programmer/Debugger User's Guide**. 2009. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/PICkit_3_User_Guide_51795A.pdf>. Acesso em: 26 out. 2013.

MICROCHIP. **DS70135C: dsPIC30F4011/4012 Data Sheet - High Performance Digital Signal Controllers**. 2005. Disponível em: <<http://ww1.microchip.com/downloads/en/devicedoc/70135c.pdf>>. Acesso em: 26 out. 2013.

MICROCHIP. **DS70265B**: dsPIC33FJ12MC201/202 Data Sheet - High Performance Digital Signal Controllers. 2007. Disponível em:
< <http://ww1.microchip.com/downloads/en/DeviceDoc/70265B.pdf> >. Acesso em: 26 out. 2013.

NETO, M. R. B; CARVALHO, P. C. M. **Geração de Energia Elétrica - Fundamentos**. 1. ed. São Paulo: Érica, 2012.

PEREIRA, J. C. **Motores e Geradores. Princípios de Funcionamento, Instalação, Operação e Manutenção de Grupos Diesel Geradores**. [2006]. Disponível em:
<<http://www.joseclaudio.eng.br/geradores/PDF/diesel1.pdf>>. Acesso em: 12 set. 2013.

TAYLOR, F. T. ***The Internal combustion engine in theory and practice***. 2. ed. Washington: The Massachusetts Institute of Technology, 1985.

THOMAZINI, D.; ALBUQUERQUE, P. U. B. **Sensores Industriais Fundamentos e Aplicações**. 4 ed. São Paulo: Érica, 2007.

TRANSPETRO. O que é o PROMEF. Disponível em: <<http://www.promef-transpetro.com.br/>> Acesso em: 02 set. 2013.

VDO. **Sensores de pressão de óleo**. [2013?]. Disponível em:
<http://www.vdo.com.br/generator/www/br/pt/vdo/main/products_solutions/aftermarket/instrumentos_sensores/hidden/tab_aft_instr_sens_press_oleo_index_pt.html>. Acesso em: 14 set. 2013.

WOODWARD. **Governing Fundamentals and Power Management - Reference Manual**. Colorado, 2011.

APÊNDICE A - Resumo das configurações e esquemático das ligações elétricas definidas para o controlador InteliDrive Lite.

Tabela 1 - Resumo das saídas digitais utilizadas no controlador InteliDrive Lite.

SAÍDAS DIGITAIS		
#	NOME	FUNÇÃO
BO1	Starter	Acionamento do motor de arranque
BO2	Fuel solenoid	Abrir a válvula de combustível
BO3	Alarm	Acionar Giroflex e sirene
BO4	Não usado	-
BO5	Não usado	-
BO6	Não usado	-
BO7	Não usado	-

Fonte: Produção do próprio autor.

Tabela 2 - Resumo das entradas digitais utilizadas no controlador InteliDrive Lite.

ENTRADAS DIGITAIS			
#	NOME	FUNÇÃO	CONTATO
BI1	Pressão Óleo	<i>Shutdown</i> para pressão do óleo menor ou igual a 0,5 bar	NO
BI2	TempÁgua	<i>Shutdown</i> para temperatura da água igual ou superior a 100 °C	NO
BI3	Emergency stop	<i>Shutdown</i> do motor	NC
BI4	Não usado	-	-
BI5	Não usado	-	-
BI6	Não usado	-	-
BI7	Não usado	-	-

Fonte: Produção do próprio autor.

Tabela 3 - Resumo das entradas analógicas utilizadas no controlador InteliDrive Lite.

ENTRADAS ANALÓGICAS		
#	NOME	PONTOS DA CURVA
AN1	Oil Pressure	(0 bar, 10 Ohm); (10 bar, 180 Ohm)
AN2	Water Temp	(120 °C, 22 Ohm); (0 °C, 3200 Ohm)
AN3	Não usado	-
AN4	Não usado	-
AN5	Não usado	-
AN6	Não usado	-
AN7	Não usado	-
AN8	Não usado	-
AN9	Não usado	-

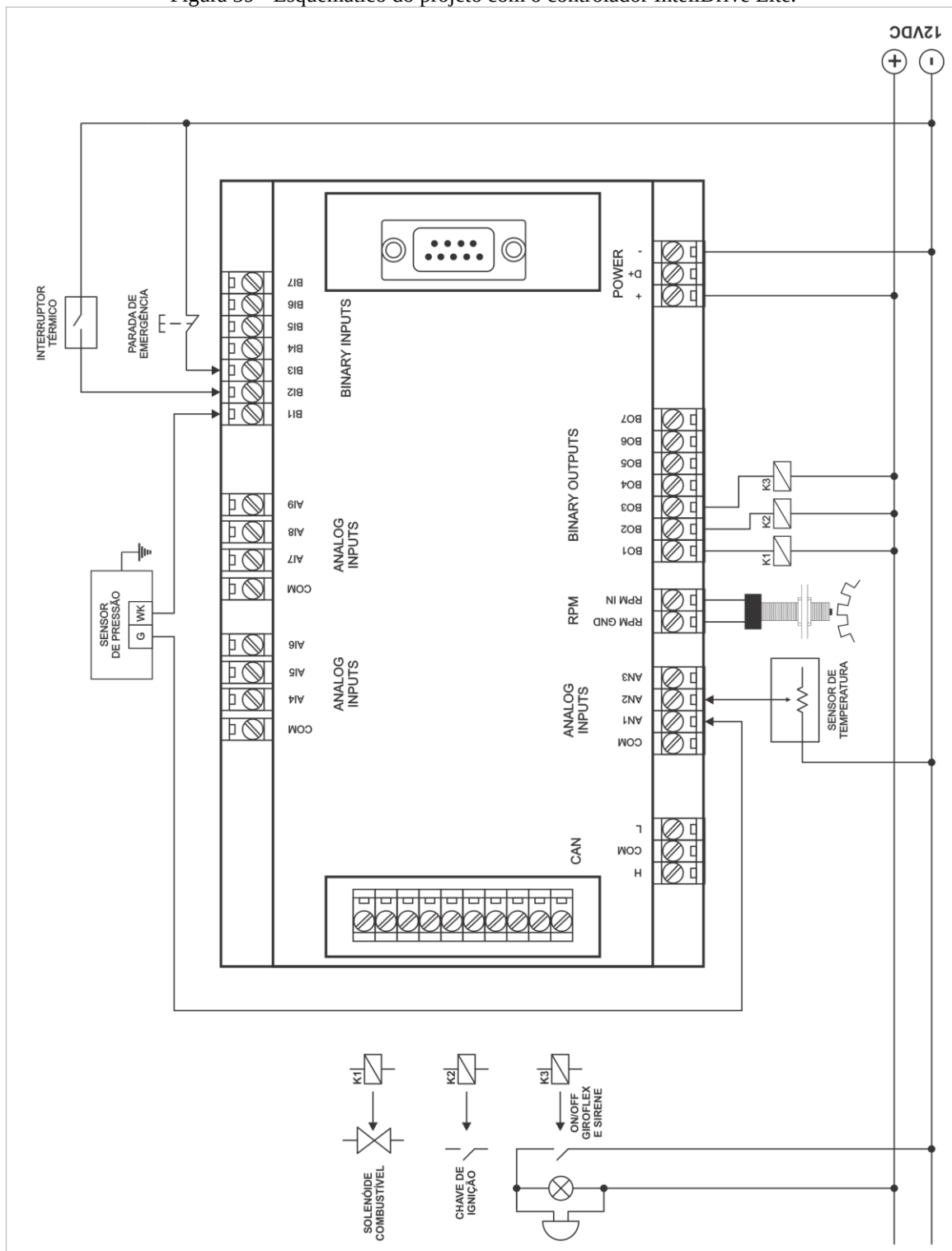
Fonte: Produção do próprio autor.

Tabela 4 - Resumo dos valores das principais variáveis do controlador InteliDrive Lite.

CONFIGURAÇÕES		
VARIÁVEL	VALOR	GRUPO
<i>Gear teeth</i>	31	<i>Basic Settings</i>
<i>RPMbyWterminal</i>	1,95	<i>Basic Settings</i>
<i>Nominal RPM</i>	1800 rpm	<i>Basic Settings</i>
<i>RPM Source</i>	<i>Pickup</i>	<i>Basic Settings</i>
<i>Starting RPM</i>	350 rpm	<i>Engine Params</i>
<i>Prestart time</i>	2 s	<i>Engine Params</i>
<i>MaxCrank time</i>	5 s	<i>Engine Params</i>
<i>Overspeed</i>	110%	<i>Engine Protect</i>
<i>Underspeed</i>	25%	<i>Engine Protect</i>
<i>Oil Pressur Wrn</i>	2,5 bar	<i>Engine Protect</i>
<i>Oil Pressur Sd</i>	2 bar	<i>Engine Protect</i>
<i>Water Temp Wrn</i>	92 °C	<i>Engine Protect</i>
<i>Water Temp Sd</i>	98 °C	<i>Engine Protect</i>
<i>Batt overvolt</i>	15,5 V	<i>Engine Protect</i>
<i>Batt undervolt</i>	8,0 V	<i>Engine Protect</i>

Fonte: Produção do próprio autor.

Figura 39 - Esquemático do projeto com o controlador IntelliDrive Lite.



Fonte: Produção do próprio autor.

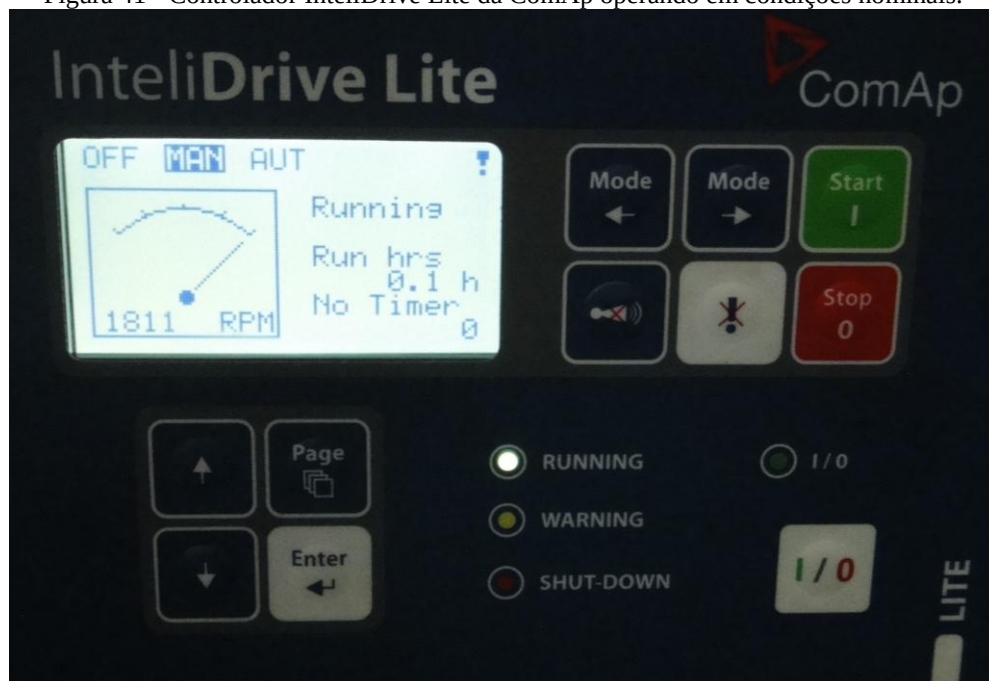
APÊNDICE B - Imagens do painel no qual foi instalado o controlador InteliDrive Lite.

Figura 40 - Painel do grupo gerador com controlador da ComAp.



Fonte: Produção do próprio autor.

Figura 41 - Controlador InteliDrive Lite da ComAp operando em condições nominais.



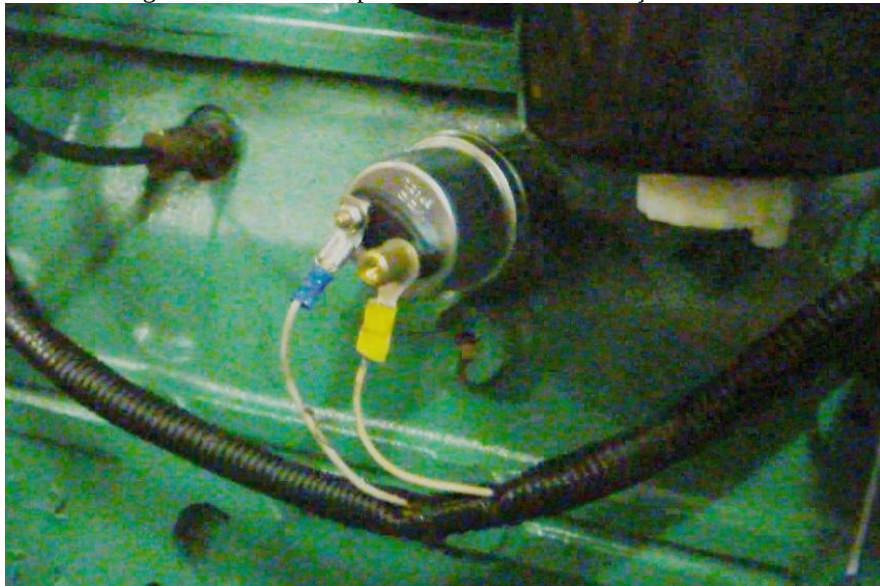
Fonte: Produção do próprio autor.

Figura 42 - *Pickup* magnético instalado na carcaça do motor.



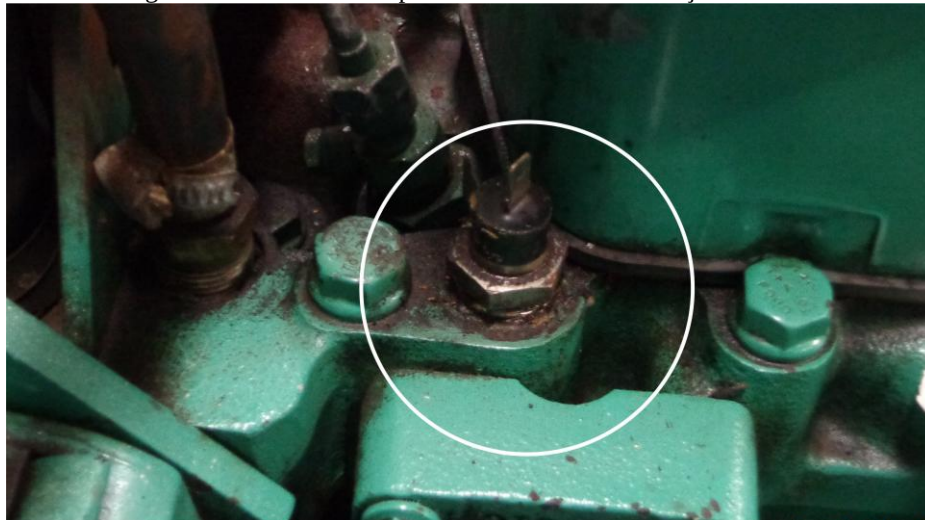
Fonte: Produção do próprio autor.

Figura 43 - Sensor de pressão instalado na carcaça do motor.



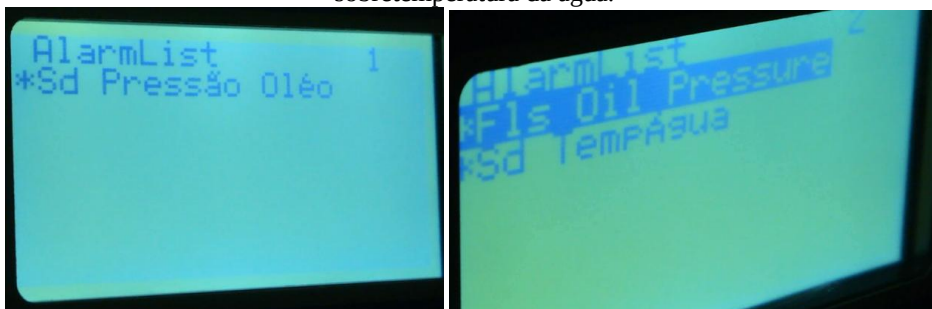
Fonte: Produção do próprio autor.

Figura 44 - Sensor de temperatura instalado na carcaça do motor.



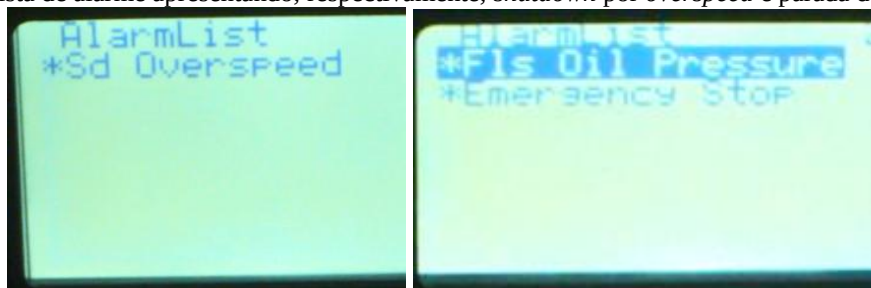
Fonte: Produção do próprio autor.

Figura 45 - Lista de alarme apresentando, respectivamente, *shutdown* por perda de pressão do óleo e sobretemperatura da água.



Fonte: Produção do próprio autor.

Figura 46 Lista de alarme apresentando, respectivamente, *shutdown* por *overspeed* e parada de emergência.



Fonte: Produção do próprio autor.

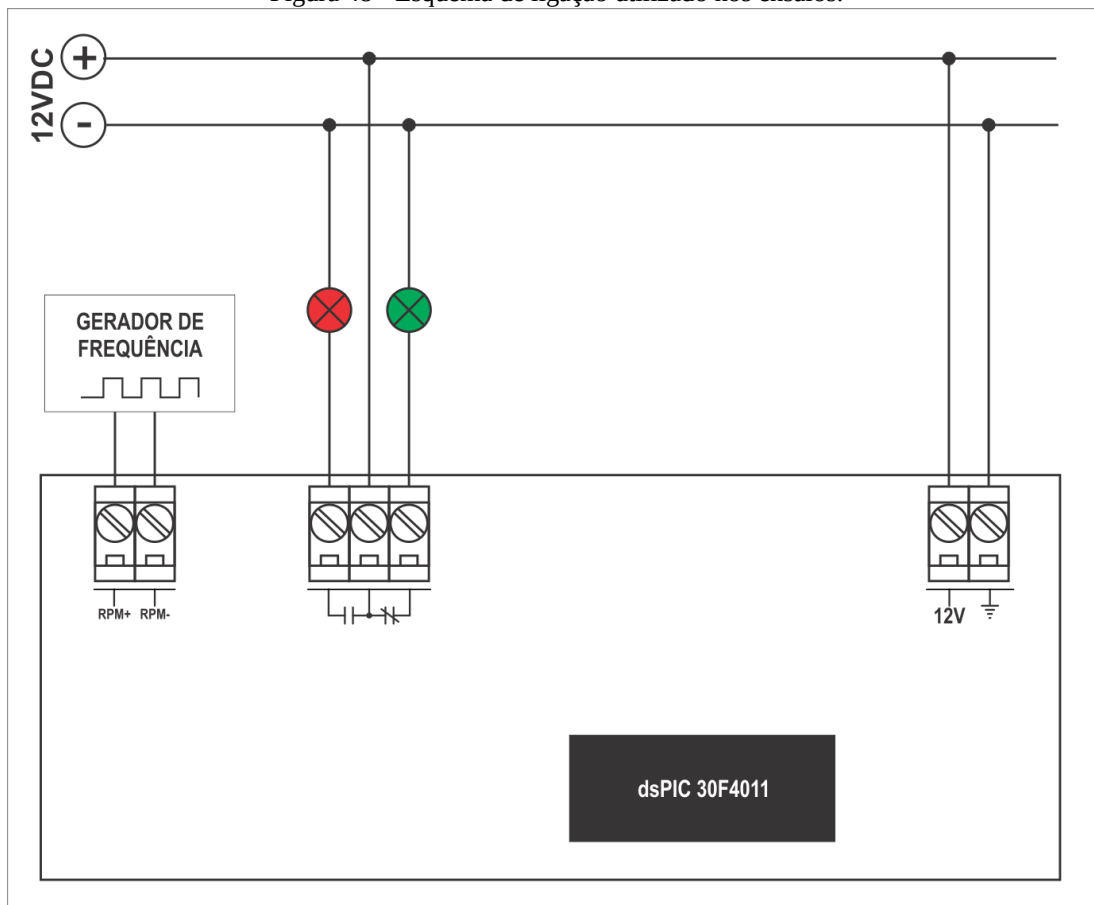
Figura 47 - *Displays* de apresentação das variáveis monitoradas pelo controlador.



Fonte: Produção do próprio autor.

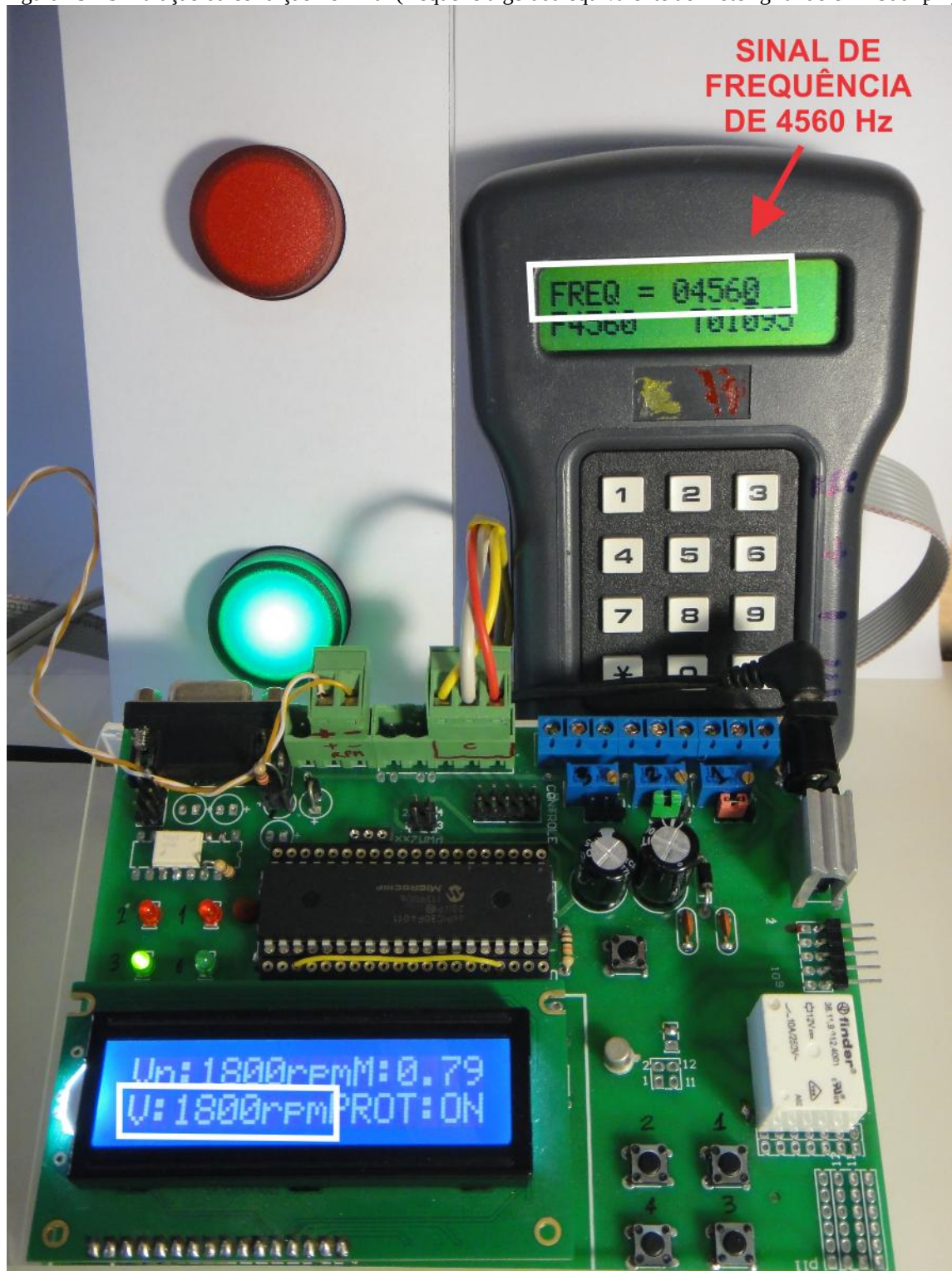
APÊNDICE C - Ensaios realizados com gerador de frequência inserindo o sinal na placa com o dsPIC30F4011.

Figura 48 - Esquema de ligação utilizado nos ensaios.



Fonte: Produção do próprio autor.

Figura 49 - Simulação da condição nominal (frequência gerada equivalente ao motor girando em 1800 rpm).



Fonte: Produção do próprio autor.

Figura 50 - Simulação do motor operando com 500 rpm.



Fonte: Produção do próprio autor.

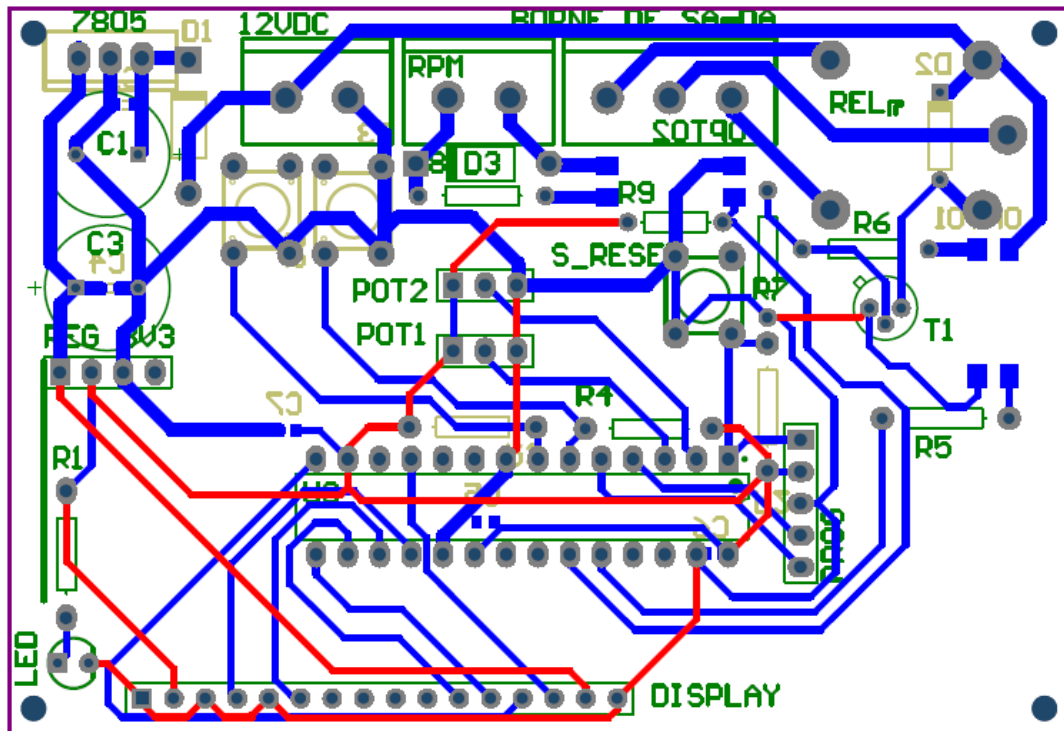
Figura 51 - Simulação de uma situação de *overspeed* de exatamente 110%.



Fonte: Produção do próprio autor.

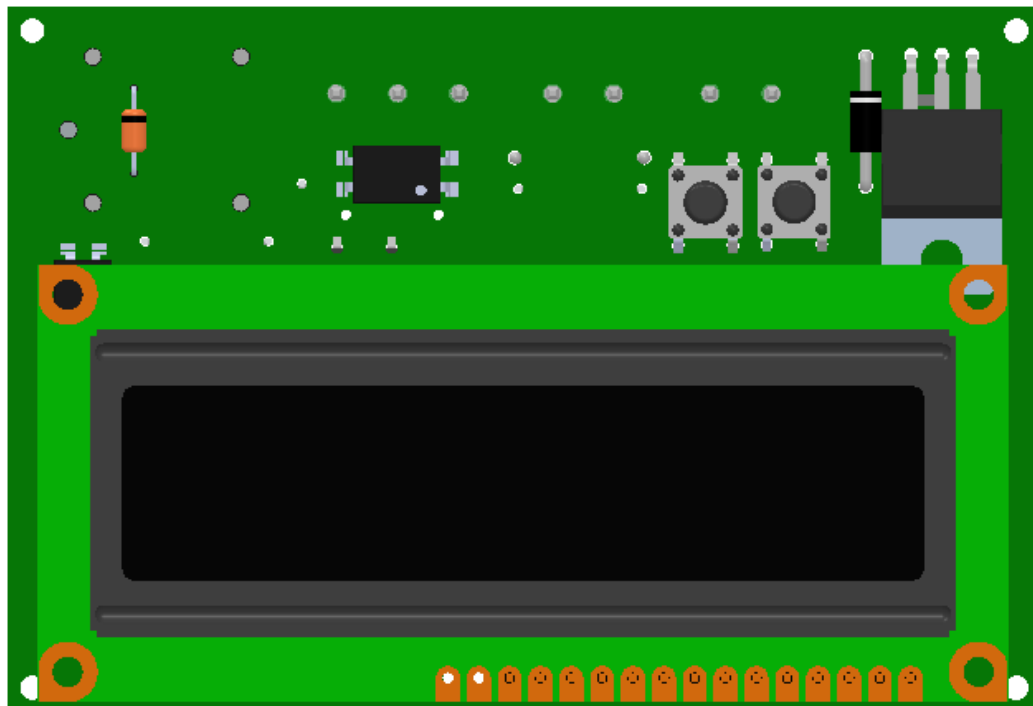
Fonte: Produção do próprio autor.

Figura 53 - *Layout* da placa de circuito impresso contendo o dsPIC33FJ12MC202 produzido no Altium Designer.



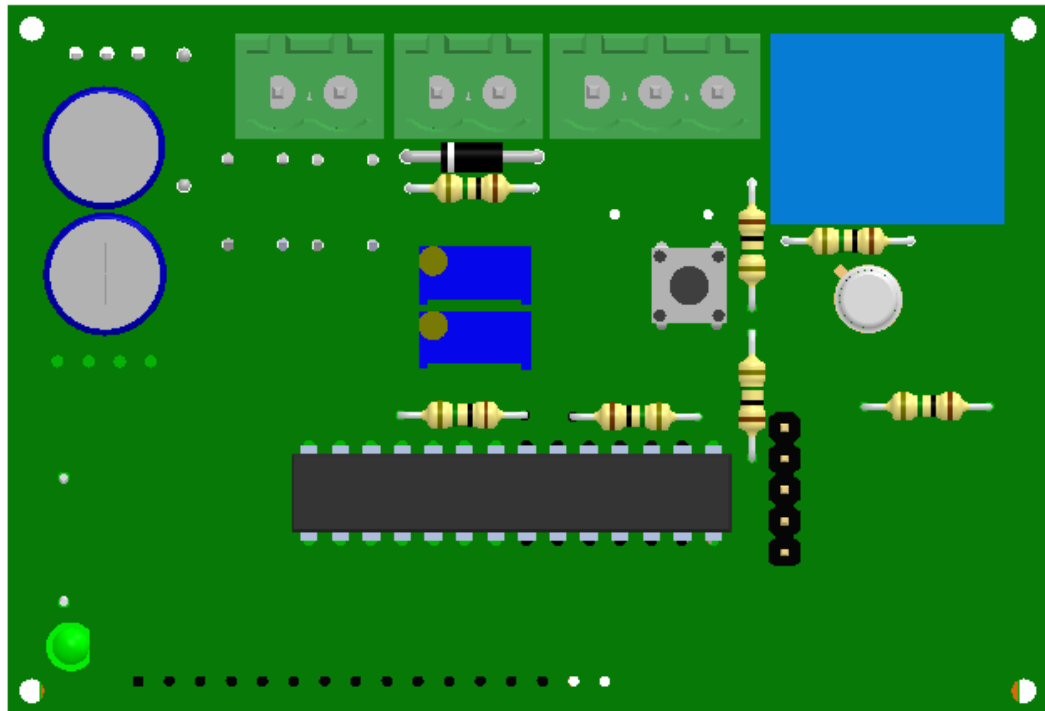
Fonte: Produção do próprio autor.

Figura 54 - Vista superior gerada através do *SolidWorks* e desenvolvida no *Altium Designer* com o dsPIC33FJ12MC202.



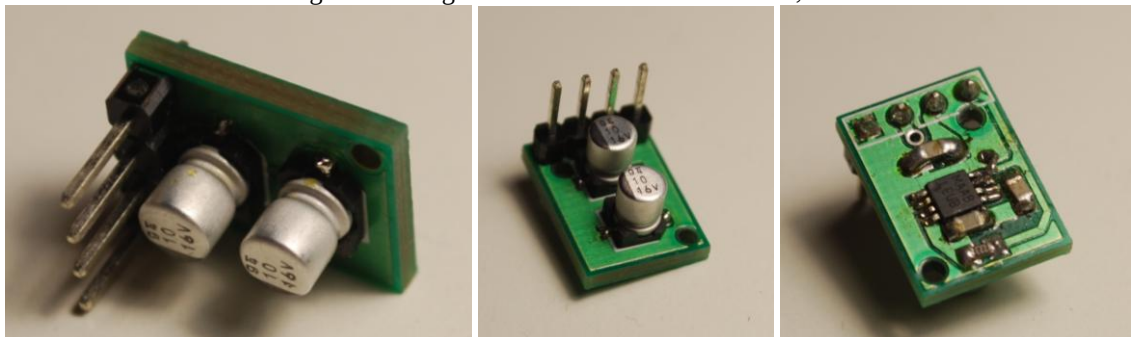
Fonte: Produção do próprio autor.

Figura 55 - Vista inferior gerada através do *SolidWorks* e desenvolvida no *Altium Designer* com o dsPIC33FJ12MC202.



Fonte: Produção do próprio autor.

Figura 56 - Regulador de tensão com saída de 3,3 V.



Fonte: Produção do próprio autor.

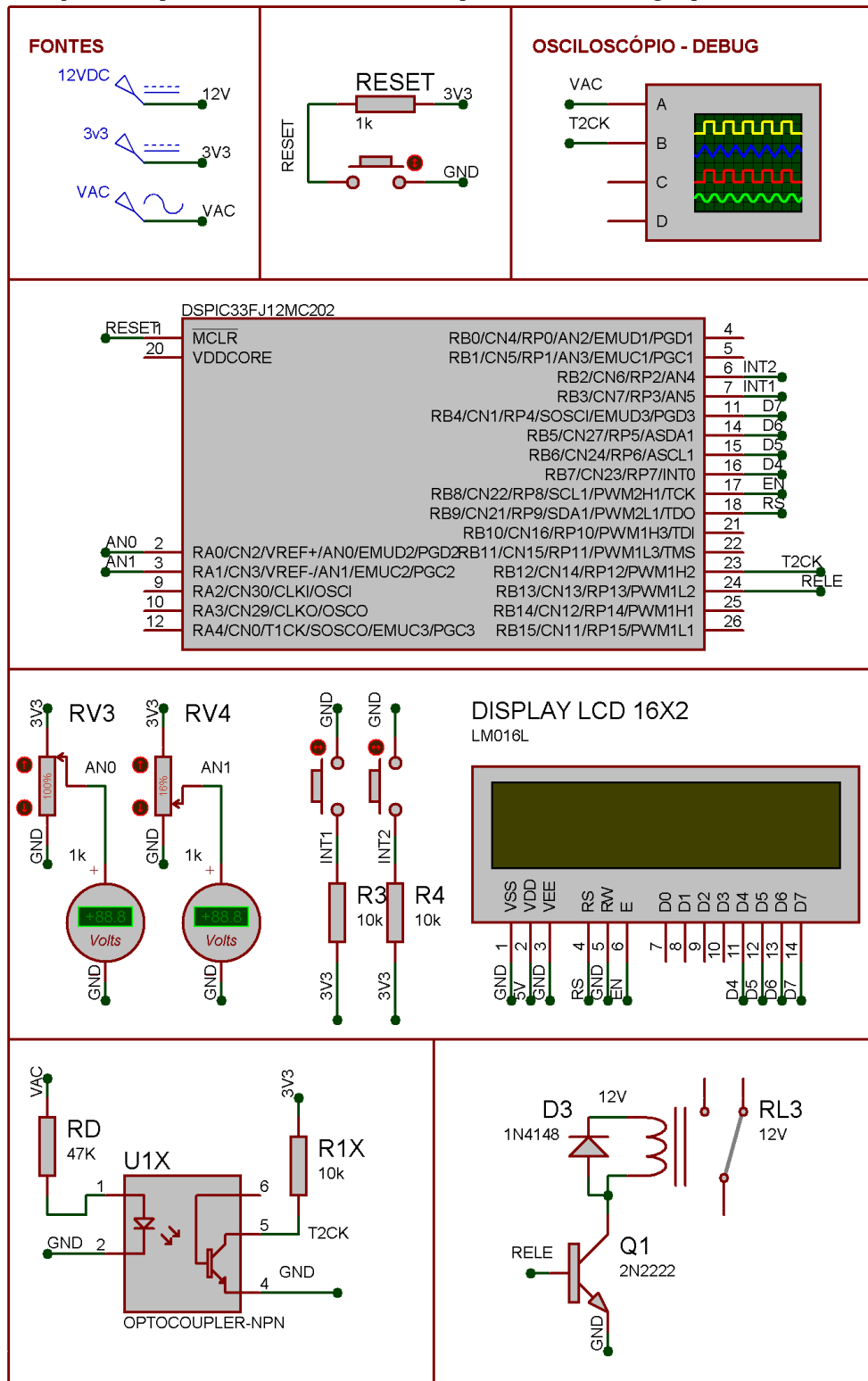
Tabela 5 - Lista de materiais necessários.

LISTA DE MATERIAIS			
ITEM	DESCRIÇÃO	QUANT.	IMAGEM
RPM	Borne de dois pinos para conexão do sensor de velocidade e para alimentação de 12 VDC	2	
7805	Regulador de tensão com saída em 5 VDC	1	
BORNE DE SAÍDA	Borne de três pinos para conexão dos pinos NAF do relé	1	
C1	Capacitor eletrolítico de 1000uF/16V	1	
C3	Capacitor eletrolítico de 470uF/16V	1	
C2, C4, C5, C6, C7	Capacitor cerâmico SMD de 100nF	5	
DISPLAY	Display LCD 16X2	1	
D1, D3	Diodo 1N400X	2	
D2	Diodo 1nN4148	1	
LED	LED 3mm verde	1	
OPTO1, OPTO2	Optoacoplador PC817	2	
POT1, POT2	Potenciômetro trimmer 25 volts 500mW 10KOhm	2	

PROG	Conector header macho de 5 vias	1	
REG 3V3	Conector header macho de 4 vias para conectar placa reguladora com saída de 3,3 VDC	1	
T1	Transistor NPN 2N2222 30V 800mA	1	
R1, R5, R6, R7, R9	Resistor 1/4W 390Ohm	5	
R2, R3, R4	Resistor 1/4W 10KOhm	3	
R8	Resistor 1/4W 100Ohm	1	
RELÉ	Relé com tensão nominal de 12 VDC, consumo de 30mA e um contato reversível (1NAF)	1	
BUTTON1, BUTTON2, RESET	Push-button	3	
MC	dsPIC33FJ12MC202 MICROCHIP	1	

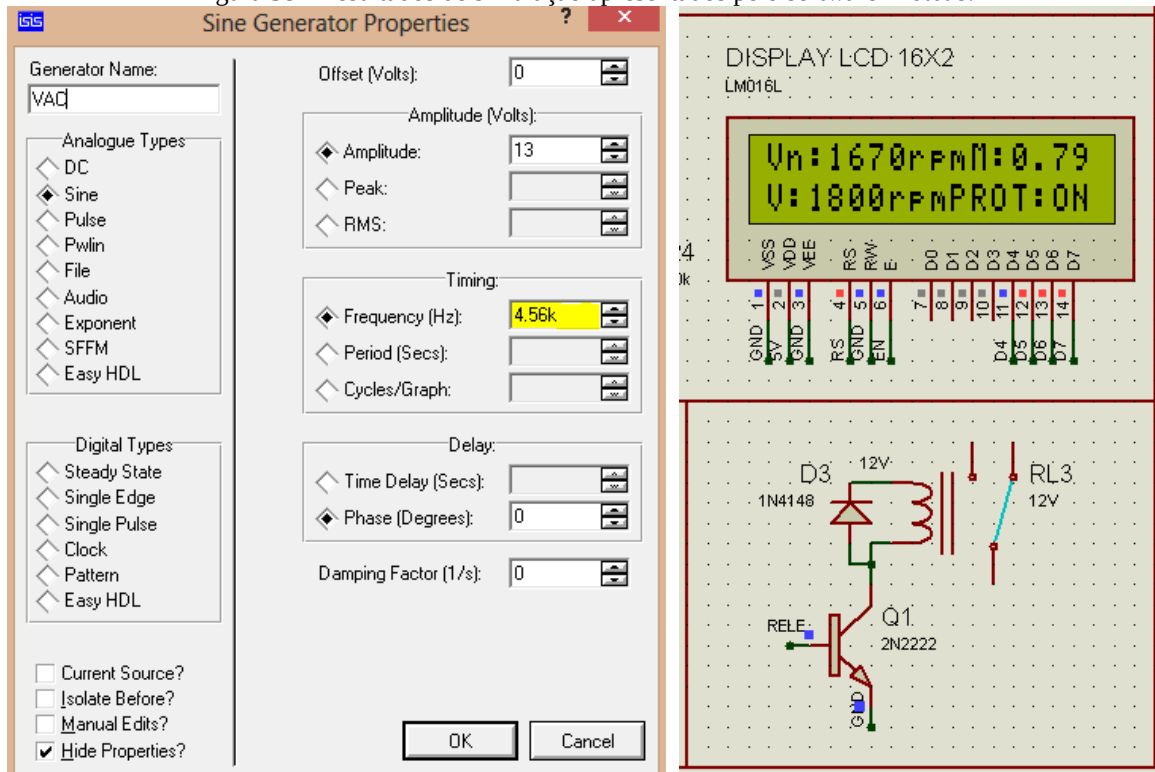
Fonte: Produção do próprio autor.

Figura 57 - Esquemático produzido no software *Proteus* para simular os códigos para o dsPIC33FJ12MC202.



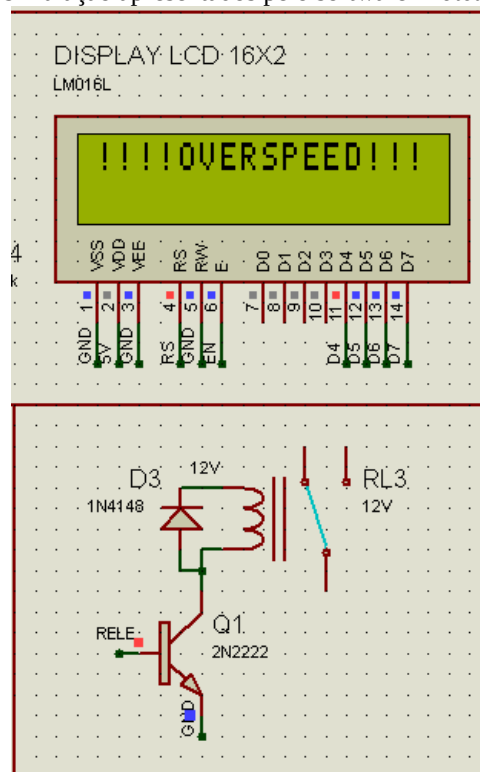
Fonte: Produção do próprio autor.

Figura 58 - Resultados de simulação apresentados pelo software *Proteus*.



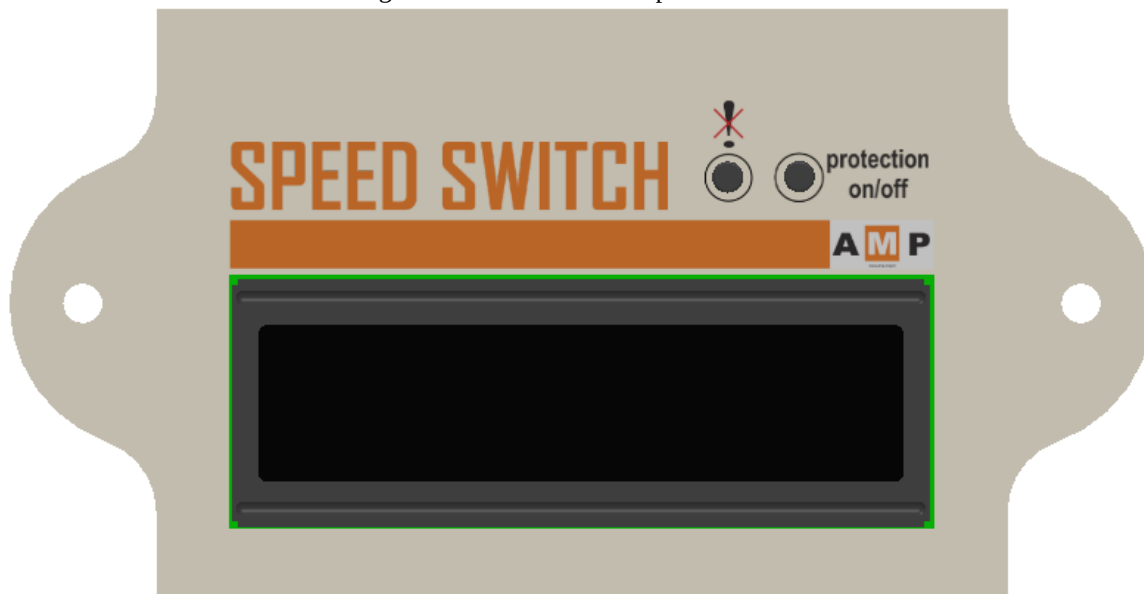
Fonte: Produção do próprio autor.

Figura 59 - Resultados de simulação apresentados pelo software *Proteus* em uma situação de falha.



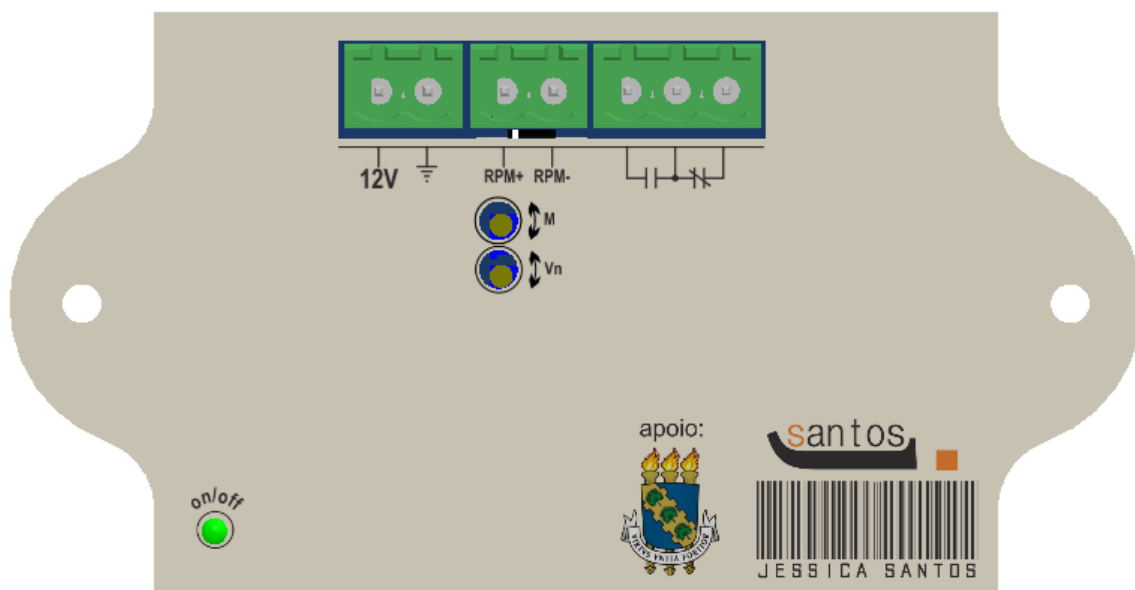
Fonte: Produção do próprio autor.

Figura 60 - Vista frontal do produto final.



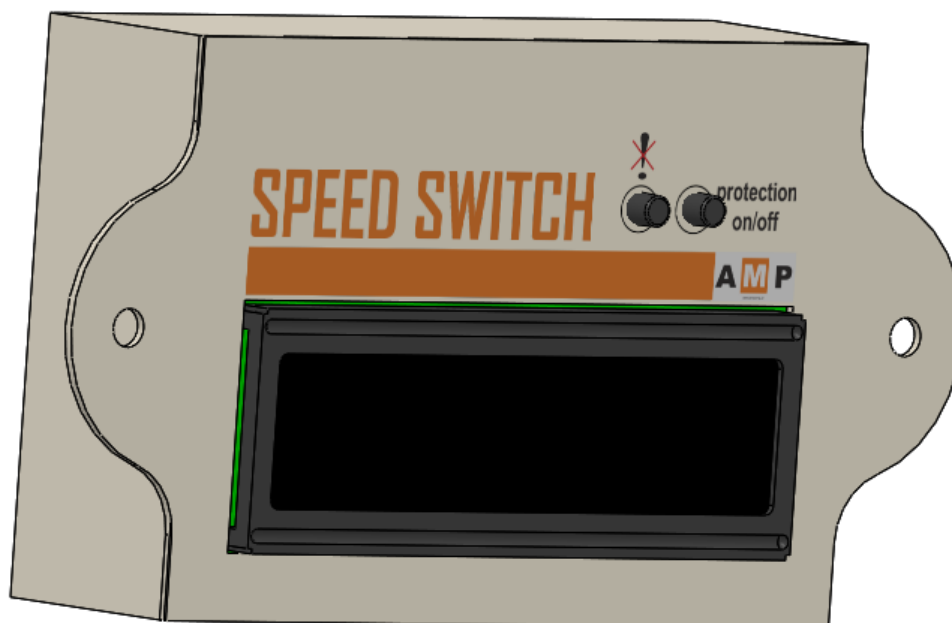
Fonte: Produção do próprio autor.

Figura 61 - Vista posterior do produto final.



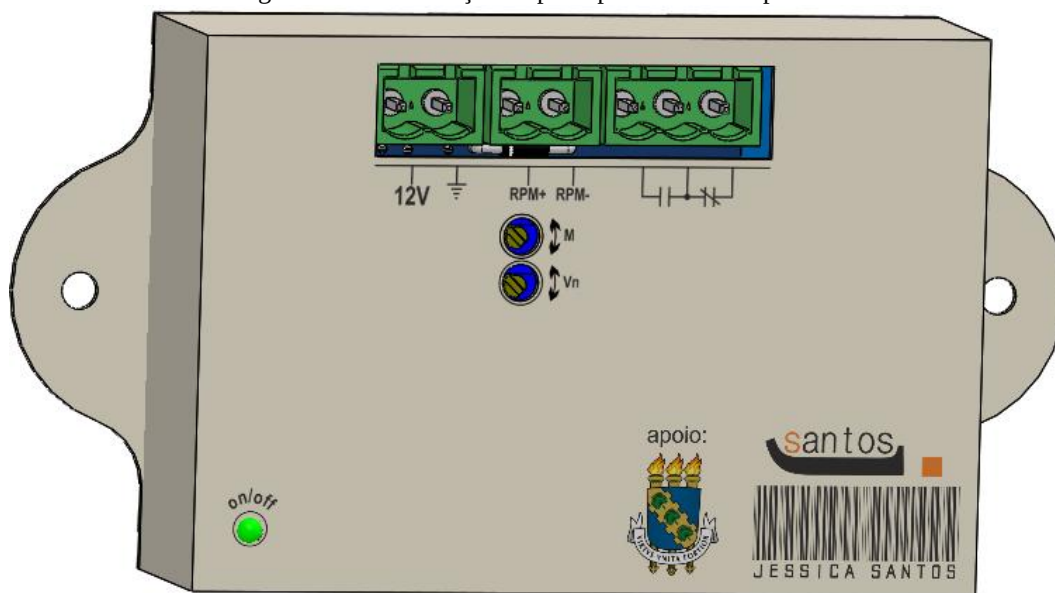
Fonte: Produção do próprio autor.

Figura 62 - Visualização do frontal do dispositivo.



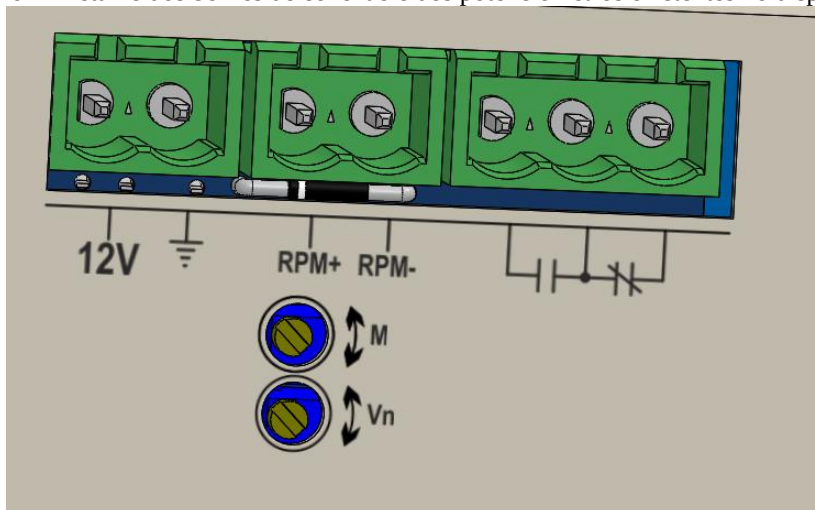
Fonte: Produção do próprio autor.

Figura 63 - Visualização da parte posterior do dispositivo.



Fonte: Produção do próprio autor.

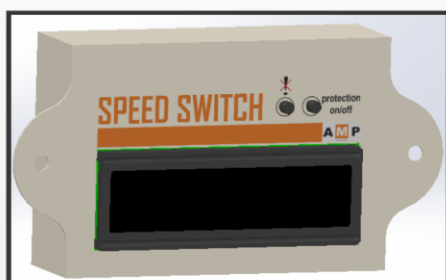
Figura 64 - Detalhe dos bornes de conexão e dos potenciômetros existentes no dispositivo.



Fonte: Produção do próprio autor.

Figura 65 - Folheto desenvolvido para o dispositivo proposto.

SPEED SWITCH



DESCRIÇÃO

Dispositivo de medição de velocidade com contatos secos que alternam seus estágios quando a velocidade atinge valores igual ou superior a 110% do valor nominal configurável.

APLICAÇÃO

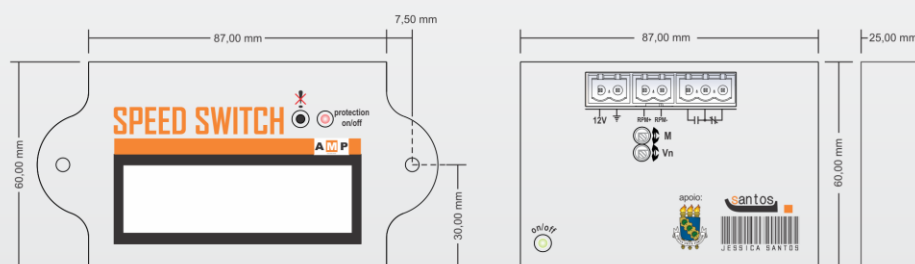
Proteção de motores causando um *shut-down* dos mesmos quando houver existência de *overspeed*.

ESPECIFICAÇÕES

Tensão nominal: 12 VDC

Relé de saída: 1NAF, 10A (250/125VAC, 30/28VDC)

DIMENSÕES

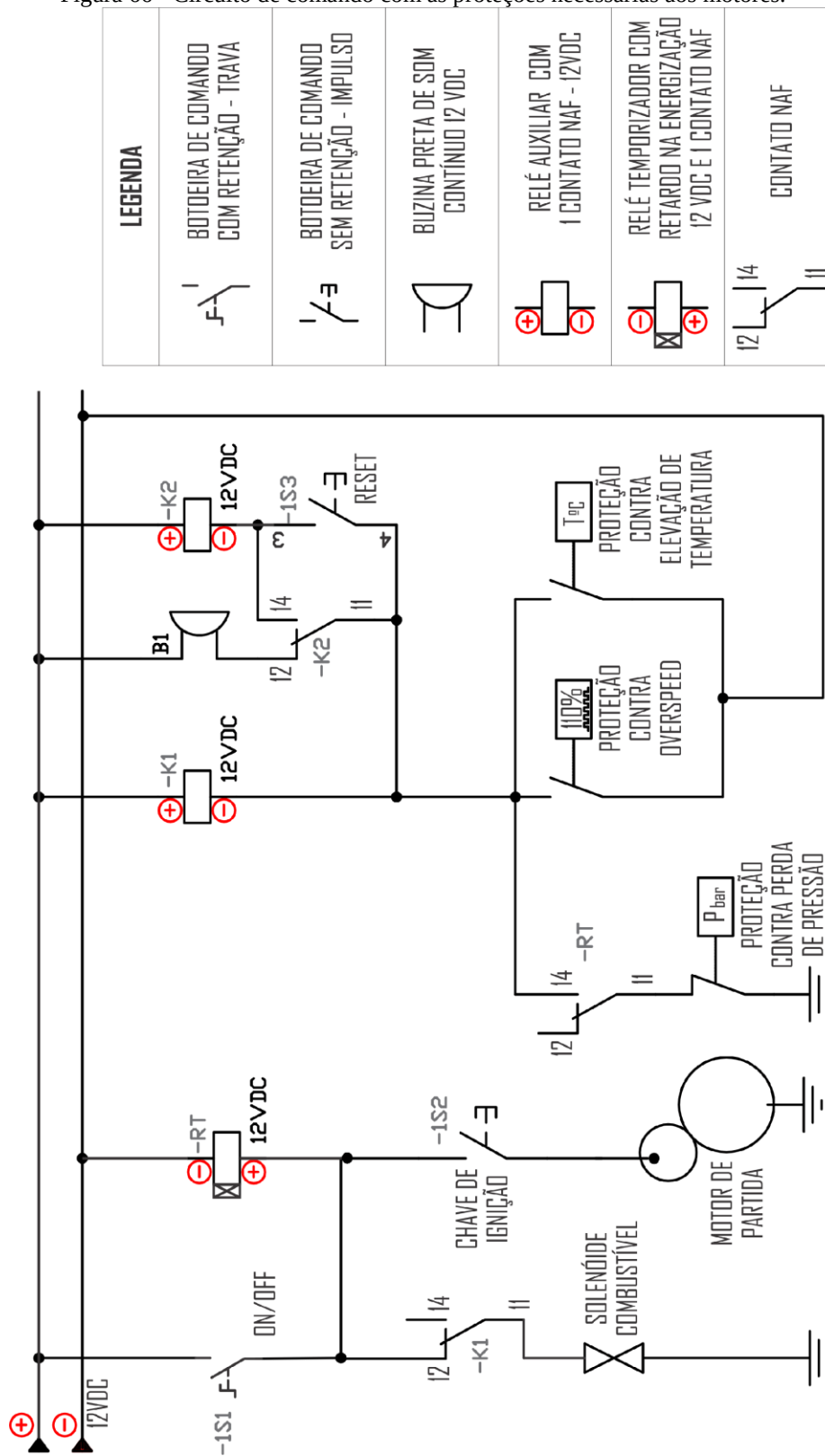


PRODUZIDO POR: Jéssica Santos Guimarães

Fonte: Produção do próprio autor.

APÊNDICE E - Exemplo de aplicação.

Figura 66 - Circuito de comando com as proteções necessárias aos motores.



APÊNDICE F - Códigos utilizados no dsPIC 30F4011.

Código 1 - Arquivo "main.h" para o dsPIC 30F4011.

```

_FOSC(CSW_FSCM_OFF & FRC_PLL4);           //CHAVEAMENTO DE CLOCK E MONITOR OFF
                                           //CLOCK INTERNO COM PLL 4X
_FWDT(WDT_OFF);                           //WATCHDOG DESABILITADO

_FBORPOR(MCLR_EN & PBOR_ON & BORV_27 & PWRT_64);
//MASTER CLEAR HABILITADO
//BROWN OUT HABILITADO
//TENSÃO DE RESET DO BROWN OUT DE 2,7V
//POWER UP TIMER HABILITADO PARA 64ms

#define Fcy 7370000
// cristal interno = 7.37MHz ---> FCY = (CRISTAL x PLL)/4 = 7.37MHz
#define Tcy 136           //nanosegundos
#define cyc 500/Tcy

//----->DEFININDO PRIORIDADES DAS INTERRUPÇÕES<-----
#define PRI_INT1    4    //INTERRUPÇÃO EXTENA 1
#define PRI_INT2    4    //INTERRUPÇÃO EXTENA 2
#define PRI_TMR1    5    //TIMER 1
#define PRI_TMR2    7    //TIMER 2
#define PRI_TMR3    1    //TIMER 3
#define PRI_AD      6    //CONVERSOR ANALÓGICO-DIGITAL

//----->Tags<-----

#define LED1 LATEbits.LATE8
#define LED2 LATCbits.LATC13
#define LED3 LATBbits.LATB8
#define LED0 LATFbits.LATF6

#define BUTT1 PORTEbits.RE1
#define BUTT2 PORTEbits.RE3
#define BUTT3 PORTEbits.RE4
#define BUTT4 PORTEbits.RE5

#define RELE LATBbits.LATB3
//----->INPUT OUTPUT<-----

void init (void)
{

    TRISBbits.TRISB0=1; //AN0
    TRISBbits.TRISB1=1; //AN1
    TRISBbits.TRISB2=1; //AN2

    TRISFbits.TRISF6=0; //LED0
    TRISEbits.TRISE8=0; //LED1
    TRISCbits.TRISC13=0; //LED2
    TRISBbits.TRISB8=0; //LED3

    TRISEbits.TRISE1=1; //BUTT1
    TRISDbits.TRISD0=1; //INT1

    TRISEbits.TRISE3=1; //BUTT2
    TRISDbits.TRISD1=1; //INT2

    TRISEbits.TRISE4=1; //BUTT3
    TRISEbits.TRISE5=1; //BUTT4

    TRISDbits.TRISD3 = 0; //RS
    TRISDbits.TRISD2 = 0; //EN
    TRISFbits.TRISF3 = 0; //D4

```

```

TRISFbits.TRISF2 = 0; //D5
TRISFbits.TRISF1 = 0; //D6
TRISFbits.TRISF0 = 0; //D7

TRISCbits.TRISC14=1; // ENTRADA DE CLOCK EXTERNO T1CK
TRISBbits.TRISB3=0; // ACIONAMENTO DO RELE

ADPCFG = 0b1111111111111100; //ANO AN1
//1 = Port pin in Digital mode, port read input
//enabled, ADC input multiplexer connected to AVSS
//0 = Port pin in Analog mode, port read input
//disabled, ADC samples pin voltage
}

//----->FLAGS<-----
unsigned int flag_0 = 0;
// FLAG PARA SINALIZAR O ESTOURO DO TIMER 2 QUE NÃO DEVE ACONTECER
unsigned int flag_EN = 1;
//FLAG PARA HABILITAR OU NÃO A PROTEÇÃO
unsigned int shutdown = 0;
//FLAG QUE SINALIZA OCORRENCIA DE SHUTDOWN POR OVERSPEED (Vmedida > Vn)
unsigned int warning = 0;
//FLAG QUE DETECTA QUANDO Vmedida É MAIOR QUE Vn

//----->VARIÁVEIS<-----
unsigned int CONTAGEM;
//variável que armazena o número de bordas de subida do sinal do pickup
//a cada intervalo de tempo (0,5s)

unsigned int Vn;
//variável que armazena o valor da velocidade nominal
//definida através do potenciometro
unsigned int Vn_AD; //variável auxiliar para o cálculo do valor de Vn
unsigned long int SOMAAD1; //variável auxiliar para o cálculo do valor de Vn

float M;
//variável que armazena o valor do múltiplo que define a velocidade real do motor
float M_AD;
//variável auxiliar para o cálculo do valor do múltiplo M
unsigned long int SOMAAD2;
//variável auxiliar para o cálculo do valor do múltiplo M

unsigned int Vlimite;
//valor limite da velocidade a partir do qual a proteção deve ser habilitada
unsigned int Vmedida; //valor instantâneo da velocidade do motor
float AJUSTEFIXO = 1.1;
//valor permitido de sobrevelocidade, sem que a proteção atue = 10%

unsigned int adn=0; //contador auxiliar
unsigned long int displayn=0; //contador auxiliar

```

Fonte: Produção do próprio autor.

Código 2 - Arquivo "timer_x.h" para o dsPIC 30F4011.

```

unsigned long int f_base = 1000;
unsigned long int m_base = 1;
unsigned long int tmr3_cont = 0;
unsigned int dj,dnop;

void config_tmr3 (long int frequencia, long int prescale, int start)
{
//----->Inicialização<---
    T3CONbits.TON = 0; //desligado
    PR3 = Fcy/(prescale*frequencia);
    TMR3 = 0;
    T3CONbits.TGATE = 0;
    T3CONbits.TCS = 0;
//----->Interrupções<-----

```

```

IPC1bits.T3IP = PRI_TMR3;
IFS0bits.T3IF = 0;
IEC0bits.T3IE = 1;

//----->Prescale<-----
if (prescale == 1)           //Atribuindo time base input prescale
    T3CONbits.TCKPS = 0b00;
else if (prescale == 8)
    T3CONbits.TCKPS = 0b01;
else if (prescale == 64)
    T3CONbits.TCKPS = 0b10;
else if (prescale == 256)
    T3CONbits.TCKPS = 0b11;
else
    T3CONbits.TCKPS = 0b00;
//----->Start<-----
    T3CONbits.TON = start;
}

void tmr3_start (int start)
{
    T3CONbits.TON = start;
}

void delay_ms (int x)
{
    unsigned long int    f_base = 1000;
    unsigned long int    m_base = 1;
    unsigned long int cont_break;

    cont_break = tmr3_cont + x*m_base;

    while (tmr3_cont != cont_break);
}

void delay_us (int x)
{
    x *= cyc;
    for (dj=0;dj<=x;dj++)
    {
        dnop++;
    }
}

void _ISR _T3Interrupt ()
{
    IFS0bits.T3IF=0; //ZERAR A FLAG
    tmr3_cont++;     //contador geral para "delay_ms()"
}

```

Fonte: Produção do próprio autor.

Código 3 - Arquivo "config_interrupts.h" para o dsPIC 30F4011.

```

void config_interrupts (void)
{
    //INTERRUPÇÃO EXTERNA INT1-----//
    _INT1IF=0;           //ZERANDO A FLAG
    _INT1IP=PRI_INT1;    //DEFININDO PRIORIDADE
    _INT1IE=1;           //HABILITANDO INTERRUPÇÃO
    //INTERRUPÇÃO EXTERNA INT2-----//
}

```

```

    _INT2IF=0;           //ZERANDO A FLAG
    _INT2IP=PRI_INT2;    //DEFININDO PRIORIDADE
    _INT2IE=1;           //HABILITANDO INTERRUPÇÃO

//TIMER 2-----//
//                                     (base de tempo de 0,5segundos)

    T2CONbits.TON=0;      //DESABILITADO
    T2CONbits.TCKPS=0b11; //PRESCALE 256
    T2CONbits.TCS=0;      //CLOCK interno
    TMR2=0x0000;          //ZERAR CONTAGEM
    PR2=14395;            //VALOR DO ESTOURO fTIMER = 2Hz ta=0.5s
    _T2IE=1;              //HABILITANDO INTERRUPÇÃO
    _T2IP=PRI_TMR2;       //DEFININDO PRIORIDADE
    _T2IF=0;              //ZERANDO A FLAG
    T2CONbits.TON=1;      //LIGAR O TIMER

//TIMER1-----//
//                                     (contador de eventos)

    T1CONbits.TON=0;      //DESABILITADO
    T1CONbits.TCKPS=0b00; //PRESCALE 1
    T1CONbits.TSYNC=0;    //CLOCK EXTERNO NÃO SINCRONIZADO
    T1CONbits.TCS=1;      //CLOCK EXTERNO NO PINO T1CK
    TMR1=0x0000;          //ZERAR CONTAGEM
    PR1=0xFFFF;          //VALOR DO ESTOURO do timer = 65635 fmax = 131070Hz

    _T1IE=1;              //HABILITANDO INTERRUPÇÃO
    _T1IP=PRI_TMR1;       //DEFININDO PRIORIDADE
    _T1IF=0;              //ZERANDO A FLAG

    T1CONbits.TON=1;      //LIGAR O TIMER

//CONVERSOR ANALÓGICO DIGITAL DE 10 BITS-----//

    _ADON=0;              //ADC off
    _FORM=0;              //FORMAT: INTEGER
    _SSRC=0b111;
// INTERNAL COUNTER ENDS SAMPLING AND STARTS CONVERSION (AUTO-CONVERT)
    _ASAM=1;              // ADC SAMPLE AUTO-START BIT
    _ALTS = 0;            //ALWAYS USES CHANNEL INPUT SELECTS FOR SAMPLE A
    _ADRC=1;              // ADC INTERNAL RC CLOCK
    _SAMC=0b111111;      // AUTO SAMPLE TIME = 31TAD (TAD=76ns)
    _VCFG=0;              //ADREF+ = AVdd; ADREF- = AVss
    _SIMSAM=1;            //SAMPLES CHANNELS SIMULTANEOUSLY

    ADCON2bits.CHPS = 0b11; //CONVERTER APENAS CANAL CH0, CH1, CH2, CH3
    _CH123NA=0;
    _CH123SA=0;          //AN0, AN1, AN2

    _SMPI=3;              //INTERRUPÇÃO A CADA 4 CONVERSÕES
    _ADIF=0;              //HABILITANDO INTERRUPÇÃO
    _ADIP=PRI_AD;         //DEFININDO PRIORIDADE
    _ADIE=1;              //ZERANDO A FLAG

    _ADON=1;              //ADC on
}

```

Fonte: Produção do próprio autor.

Código 4 - Arquivo "interrupts.h" para o dsPIC 30F4011.

```

//TIMER 2-----//
//                                     (base de tempo de 0,5segundos)

void _ISR _T2Interrupt(void)
{

```

```

    _T2IF=0;

    CONTAGEM=TMR1;
    TMR1=0;

    M=M_AD/1023;
    M=M*1.8;
    M=M+0.6;

    Vn=(int) (Vn_AD/10)*10+1000;

    Vmedida=CONTAGEM*M;
    M=M*100;
    Vlimite=Vn*AJUSTEFIXO;

    if(flag_EN)
    {
        if(Vmedida>Vn)
        {
            warning=1;
            if(Vmedida>=Vlimite)    {shutdown=1;}
        }
        else {warning=0;}
    }
}

//TIMER1-----//
//                                     (contator de eventos)
void _ISR _T1Interrupt(void)
{
    IFS0bits.T1IF=0;
    LED0=1; //SINALIZANDO A FALHA, PORQUE NUNCA A T1INT É PRA SER CHAMADA
}

//INTERRUPÇÃO EXTERNA INT1-----//
//                                     (botão 1 = ON/OFF PROTEÇÃO)
void _ISR _INT1Interrupt (void)
{
    _INT1IF=0;
    flag_EN=!flag_EN;
}

//INTERRUPÇÃO EXTERNA INT2-----//
//                                     (botão 2 = FAULT RESET)

void _ISR _INT2Interrupt (void)
{
    _INT2IF=0;
    shutdown=0;
}

//CONVERSOR ANALÓGICO DIGITAL DE 10 BITS-----//
void _ISR _ADCInterrupt(void)
{
    _ADIF=0;

    SOMAAD1+=ADCBUF1;
    SOMAAD2+=ADCBUF2;
    adn++;
    if (adn >= 23)
    {
        adn=0;
        Vn_AD=SOMAAD1/23;
        SOMAAD1=0;

        M_AD=SOMAAD2/23;;
    }
}

```

```

        SOMAAD2=0;
    }
}

```

Fonte: Produção do próprio autor.

Código 5 - Arquivo "LCD.h" para o dsPIC 30F4011.

```

//Programa: Biblioteca de escrita em LCD por 4 vias

//Esse código possui as funções básicas para escrita em um display LCD
//funcionando corretamente em um PIC 16F877a. Inclui funções para escrita de
//caractere e string...não realiza operações de leitura, nem está incluso nas
//rotinas a habilitação do modo leitura. Logo, o pino Read/Write (R/W) do LCD
//deve ser sempre aterrado (em hardware).

//Funciona no modo 4 vias, para 'economizar' pinos do microcontrolador.
//Para isso, as rotinas de configuração configuram o display para esperar sempre
//dois nibbles, e então mostrar o char recebido...

//-----//
//          Funções da biblioteca:          //
//-----//
//-->ENVIA_NIBBLE (X);   , onde x é um byte
//-->ENVIA_BYTE (Y,X);   , onde x é o byte enviado, e y especifica o tipo de dado
//                        (comando ou caractere);
//-->ESCREVE_LCD(X);     , onde x é um byte e será interpretado pelo LCD como
//                        caractere;
//-->COMANDO_LCD(X);     , onde x é um byte e será interpretado pelo LCD como
//                        comando;
//-->LIMPA_LCD();        , a função não recebe parâmetros, e limpa a tela do LCD;
//-->CONFIG_LCD();       , sem parâmetros, executa as operações padrões de
//                        configuração do LCD;
//-->FRASE(X);           , onde x é um vetor char, e todos os caracteres desse vetor
//                        são enviados ao LCD;
//-->PRINTF(FUNC,"STRING"); , onde func é qualquer função de saída de byte (das
//                        anteriores).
//-----//

//void ESCRIVE_LCD (void);

//#include <stdio.h>   //NECESSÁRIO PARA STRLEN();

#define LINE_FEED 0xC0
#define LF 0xC0
#define CR 0x10

//DEFINIÇÕES DE PINOS
#define LCD_RS LATDbits.LATD3 // DEFININDO PINOS DE CONTROLE...
#define LCD_EN LATDbits.LATD2
#define LCD_D4 LATFbits.LATF3 // E DAS 4 VIAS.
#define LCD_D5 LATFbits.LATF2
#define LCD_D6 LATFbits.LATF1
#define LCD_D7 LATFbits.LATF0

#define RS_TRIS TRISDbits.TRISD3 = 0
#define EN_TRIS TRISDbits.TRISD2 = 0
#define D4_TRIS TRISFbits.TRISF3 = 0
#define D5_TRIS TRISFbits.TRISF2 = 0
#define D6_TRIS TRISFbits.TRISF1 = 0
#define D7_TRIS TRISFbits.TRISF0 = 0

//USO DE SAIDAS FIXAS, PARA OTIMIZAR A RAM
//#USE FIXED_IO(D_OUTPUTS = PIN_D0, PIN_D1, PIN_D2, PIN_D3, PIN_D4, PIN_D5, PIN_D6,
//PIN_D7)

//#DEFINE PORTF = 0x08

```

```

//-----
// ROTINAS DO LCD...
//-----

//-->ENVIA UM NIBBLE (ESTAMOS EM SISTEMA DE 4 VIAS)<--
void ENVIA_NIBBLE (char DADO)
{
    if (DADO & 0x01) LCD_D4 = 1; else LCD_D4 = 0;    //CARREGANDO OS PINOS RD4 A RD7
    if (DADO & 0x02) LCD_D5 = 1; else LCD_D5 = 0;    //NIBBLE. OS 4 BITS MENOS
    if (DADO & 0x04) LCD_D6 = 1; else LCD_D6 = 0;
    if (DADO & 0x08) LCD_D7 = 1; else LCD_D7 = 0;

    LCD_EN = 1;                // HABILITA ENTRADAS NO LCD
    delay_us(1);               //AGUARDA
    LCD_EN = 0;                // DESABILITA...
}

//-->ENVIA UM BYTE (DOIS NIBBLES). O BYTE PODE SER UM COMANDO OU CARACTERE<--
//SE MODO = 0, ENTÃO O DADO SERÁ INTERPRETADO COMO COMANDO
//SE MODO = 1, ENTÃO O DADO SERÁ INTERPRETADO COMO CARACTERE
void ENVIA_BYTE (int MODO, char DADO)
{
    if (MODO == 1) LCD_RS = 1; else LCD_RS = 0;    //CONFIGURA O MODO...
    delay_us(50);                //AGUARDA
    LCD_EN = 0;                //DESABILITA ENTRADA
    ENVIA_NIBBLE (DADO>>4);    //ENVIA A PARTE ALTA (ROTATE_RIGHT 4 VEZES)
    ENVIA_NIBBLE (DADO);        //ZERA A ALTA E ENVIA A BAIXA...
    delay_us(40);                //AGUARDA
}

//-->SIMPLIFICACANDO O ENVIO...NESSE CASO, SÓ SE ENVIA CARACTERE<--
void ESCRIVE_LCD (char C)
{
    ENVIA_BYTE(1,C);
}

//-->E NESSE CASO, SÓ ENVIA COMANDOS<--
void COMANDO_LCD (char C)
{
    ENVIA_BYTE(0,C);
}

//-->COMANDO PARA LIMPAR A TELA<--
void LIMPA_LCD()
{
    ENVIA_BYTE (0,0x01); // COMANDO CLEAR DO DISPLAY
    delay_ms (2);
}

void ESCRIVE_NUM (int num)
{
    ESCRIVE_LCD(num + 48);
}

void LCD_CR () //cariage return
{
    COMANDO_LCD (CR);
}

void LCD_BS () //backspace
{
    COMANDO_LCD (CR);
    ESCRIVE_LCD (' ');
    COMANDO_LCD (CR);
}

```

```

//-->ROTINA DE CONFIGURAÇÃO<--
void CONFIG_LCD()
{
    RS_TRIS;      //DEFININDO SAIDAS DE ESCRITA
    EN_TRIS;
    D4_TRIS;
    D5_TRIS;
    D6_TRIS;
    D7_TRIS;

    LCD_D4 = 0;
    LCD_D5 = 0;
    LCD_D6 = 0;
    LCD_D7 = 0;
    LCD_RS = 0;      //PREPARA ESCRITA DE COMANDO DE CONFIGURAÇÃO
    LCD_EN = 0;
    delay_ms(15);
    ENVIA_NIBBLE(0x03); //INICIALIZA DISPLAY
    delay_ms(5);
    ENVIA_NIBBLE(0x02); //COMANDO 'CURSOR HOME' DO DISPLAY
    delay_ms(1);
    ENVIA_BYTE(0,0x28); //CONFIGURA O LCD PARA OPERAR EM 4 BITS, EM DUAS LINHAS.
    ENVIA_BYTE(0,0x0c); //DISPLAY LIGADO E SEM CURSOR
    LIMPA_LCD(); //CLEAR
    ENVIA_BYTE(0,0x06); //OS DADOS DESLOCAM PARA A DIREITA
}
//-----

void FRASE (char *STRING)
{
    int LEN;
    int I;
    LEN = strlen(STRING);
    I = LEN;
    while (LEN)
    {
        ESCRIVE_LCD(STRING[I - LEN]);
        LEN--;
    }
}

//-----TESTANTOOO
void EN_LCD (int n, char ndig, char mode, char punkt)
{
    //n = numero a ser escrito
    //ndig = numero de digitos
    //mode = operação com sinal (+ e -) ou sem sinal

    unsigned int i,r,num;

    if(mode==1){
        if(n<0){
            ESCRIVE_LCD('-');
            num = (unsigned int) (-n);
        }
        else{
            ESCRIVE_LCD('+');
            num = (unsigned int)n;
        }
    }
    else{
        num=(unsigned int)n;
    }

    r = num;

```

```

        i = (unsigned int)(r / (unsigned int)(10000));
        if(ndig==5) ESCRIVE_LCD(i+48);
        if(punkt==4) ESCRIVE_LCD('.');
        r = r - (unsigned int)(i*(unsigned int)10000);
        i = r / (1000L);
        if(ndig>3) ESCRIVE_LCD(i+48);
        if(punkt==3) ESCRIVE_LCD('.');
        r = r - (unsigned int)(i*(unsigned int)1000);
        i = r / (unsigned int)100L;
        if(ndig>2) ESCRIVE_LCD(i+48);
        if(punkt==2) ESCRIVE_LCD('.');
        r = r - (unsigned int)(i*(unsigned int)100);
        i = r / 10;
        if(ndig>1) ESCRIVE_LCD(i+48);
        if(punkt==1) ESCRIVE_LCD('.');
        r = r - (i*(unsigned int)10);
        ESCRIVE_LCD(r+48);
    }

void linha_lcd(int linha)
{
    if (linha == 1)
        COMANDO_LCD(0x80);
    else if (linha == 2)
        COMANDO_LCD(0xC0);
}

void posicao_lcd (int linha, int coluna)
{
    if (linha == 1)
        COMANDO_LCD(0x7F+coluna);
    if (linha == 2)
        COMANDO_LCD(0xBF+coluna);
}

```

Fonte: Produção do próprio autor.

Código 6 - Arquivo "main.c" para o dsPIC 30F4011.

```

#include <p30f4011.h>

#include "main.h"
#include "config_interrupts.h"
#include "interrupts.h"
#include "timer_x.h"
#include "LCD.h"
#include <stdio.h>

int main( )
{
    init ();
    config_interrupts();
    config_tmr3(f_base,1,1);

    LED0=0;
    LED1=0;
    LED2=0;
    LED3=0;

    RELE=0;

    CONFIG_LCD();
    delay_ms(10);

    while(1)
    {
        displayn++;
    }
}

```

```

LED3=flag_EN;
LED2=shutdown;
LED1=warning;

if(shutdown)
{
    RELE=1;
    LIMPA_LCD();
    delay_ms(500);
    posicao_lcd(1,1);
    FRASE ("!!!OVERSPEED!!!");
    delay_ms(500);
}
else
{
    RELE=0;
    LED1=0;
    if (displayn>100000)
    {
        displayn=0;

        posicao_lcd(1,1);
        FRASE ("Vn:");
        posicao_lcd(1,4);
        EN_LCD(Vn,4,0,0);
        posicao_lcd(1,8);
        FRASE ("rpm");

        posicao_lcd(2,1);
        FRASE ("V:");
        posicao_lcd(2,3);
        EN_LCD(Vmedida,4,0,0);
        posicao_lcd(2,7);
        FRASE ("rpm");

        posicao_lcd(1,11);
        FRASE ("M:");
        posicao_lcd(1,13);
        EN_LCD (M, 3, 0, 2);

        posicao_lcd(2,10);
        FRASE ("PROT:");
        if(flag_EN) {FRASE ("ON");}
        else {FRASE ("OFF");}
    }
}

} //FIM DO WHILE
} //FIM DA FUNÇÃO MAIN

```

Fonte: Produção do próprio autor.

APÊNDICE G - Códigos utilizados no dsPIC33FJ12MC202.

Código 7 - Arquivo "main.h" para o dsPIC33FJ12MC202.

```

_FOSCSEL(FNOSC_FRC)

#define Fcy 3685000 // Fcy = Fosc/2= FRC/2 = 7,37MHz/2 = 3,685MHz
#define Tcy 271 // 1/Fcy (nanosegundos)
#define cyc 500/Tcy

//----->DEFININDO PRIORIDADES DAS INTERRUPTOES<-----
#define PRI_INT1 4 //INTERRUPÇÃO EXTENA 1
#define PRI_INT2 4 //INTERRUPÇÃO EXTENA 2
#define PRI_TMR1 5 //TIMER 1
#define PRI_TMR2 7 //TIMER 2
#define PRI_TMR3 1 //TIMER 3
#define PRI_AD 6 //CONVERSOR ANALÓGICO-DIGITAL

//----->Tags<-----
#define RELE LATBbits.LATB13
//----->INPUT OUTPUT<-----

void init (void)
{

TRISAbits.TRISA0=1; //AN0
TRISAbits.TRISA1=1; //AN1

TRISBbits.TRISB2=1; //AN2
TRISBbits.TRISB3=1; //INT1
TRISBbits.TRISB4=0; //D7
TRISBbits.TRISB5=0; //D6
TRISBbits.TRISB6=0; //D5
TRISBbits.TRISB7=0; //D4
TRISBbits.TRISB8=0; //EN
TRISBbits.TRISB9=0; //RS

TRISBbits.TRISB12=1; // ENTRADA DE CLOCK EXTERNO T2CK
TRISBbits.TRISB13=0; // ACIONAMENTO DO RELE

AD1PCFGL=0b1111111111111100; //ANO AN1 AN2
//1 = Port pin in Digital mode, port read input enabled,
//ADC input multiplexer connected to AVSS
//0 = Port pin in Analog mode, port read input disabled,
//ADC samples pin voltage

RPINR0bits.INT1R=3; //INTERRUPÇÃO EXTERNA 1 MAPEADA NA PORTA RP3
RPINR1bits.INT2R=2; //INTERRUPÇÃO EXTERNA 2 MAPEADA NA PORTA RP2

RPINR3bits.T2CKR=12; //ENTRADA DE CLOCK EXTERNO PARA TIMER 2 = QEA

}

//----->FLAGS<-----
unsigned int flag_0 = 0;
// FLAG PARA SINALIZAR O ESTOURO DO TIMER 2 QUE NÃO DEVE ACONTECER
unsigned int flag_EN = 1;
//FLAG PARA HABILITAR OU NÃO A PROTEÇÃO
unsigned int shutdown = 0;
//FLAG QUE SINALIZA OCORRENCIA DE SHUTDOWN POR OVERSPEED (Vmedida > Vn)
unsigned int warning = 0;
//FLAG QUE DETECTA QUANDO Vmedida É MAIOR QUE Vn

//----->VARIÁVEIS<-----
unsigned int CONTAGEM; //variável que armazena o número de bordas de subida do
//sinal do sensor de velocidade
//a cada intervalo de tempo (0,5s)
unsigned int Vn; //variável que armazena o valor da velocidade nominal

```

```

definida através do potenciometro
unsigned int Vn_AD;           //variável auxiliar para o cálculo do valor de Vn
unsigned long int SOMAAD1;     //variável auxiliar para o cálculo do valor de Vn

float M;                      //variável que armazena o valor do múltiplo que define
a velocidade real do motor
float M_AD;                   //variável auxiliar para o cálculo do valor do múltiplo
M
unsigned long int SOMAAD2;     //variável auxiliar para o cálculo do valor do múltiplo
M

unsigned int Vlimite;          //valor limite da velocidade a partir do qual a
proteção deve ser habilitada
unsigned int Vmedida;          //valor instantâneo da velocidade do motor
float AJUSTEFIXO = 1.1;        //valor permitido de sobrevelocidade, sem que a
proteção atue = 10%

unsigned int adn=0;            //contador auxiliar
unsigned long int displayn=0;  //contador auxiliar

```

Fonte: Produção do próprio autor.

Código 8 - Arquivo "config_interrupts.h" para o dsPIC33FJ12MC202.

```

void config_interrupts (void)
{
//INTERRUPÇÃO EXTERNA INT1-----//
  _INT1IF=0;           //ZERANDO A FLAG
  _INT1IP=PRI_INT1;    //DEFININDO PRIORIDADE
  _INT1IE=1;           //HABILITANDO INTERRUPÇÃO

//INTERRUPÇÃO EXTERNA INT2-----//
  _INT2IF=0;           //ZERANDO A FLAG
  _INT2IP=PRI_INT2;    //DEFININDO PRIORIDADE
  _INT2IE=1;           //HABILITANDO INTERRUPÇÃO

//TIMER 1-----//
//                               (base de tempo de 0,5segundos)

  T1CONbits.TON=0;      //DESABILITADO
  T1CONbits.TCS=0;      //CLOCK INTERNO FCY=3685000
  T1CONbits.TCKPS=0b11; //PRESCALE = 256
  T1CONbits.TGATE=0;
  TMR1=0;               //ZERAR CONTAGEM
  PR1=7196;             //VALOR DO ESTOURO fTIMER = 2Hz ta=0.5s

  IFS0bits.T1IF=0;      //ZERAR A FLAG
  IPC0bits.T1IP=PRI_TMR1; //PRIORIDADE
  IEC0bits.T1IE=1;      //HABILITAR INTERRUPÇÃO
  T1CONbits.TON=1;      //LIGAR O TIMER

//TIMER2-----//
//                               (contador de eventos)

  T2CONbits.TON=0;      //DESABILITADO
  T2CONbits.T32=0;
  T2CONbits.TCS=1;      //CLOCK EXTERNO no pino T2CK
  T2CONbits.TCKPS=0b00; //PRESCALE 1
  TMR2=0x0000;          //ZERAR CONTAGEM
  PR2=0xFFFF;           //VALOR DO ESTOURO do timer = 65635
  _T2IF=0;              //ZERAR A FLAG
  _T2IP=PRI_TMR2;       //PRIORIDADE
  _T2IE=1;              //HABILITAR INTERRUPÇÃO
  T2CONbits.TON=1;      //LIGAR O TIMER

//CONVERSOR ANALÓGICO DIGITAL DE 10 BITS-----//
  ADON=0;               //ADC off

```

```

_AD12B=0;           //10bits
_FORM=0;            //FORMAT: INTEGER
_SSRC=0b111;       // INTERNAL COUNTER ENDS SAMPLING AND STARTS CONVERSION
(AUTO-CONVERT)
_ASAM=1;           // ADC SAMPLE AUTO-START BIT
_ALTS = 0;         //ALWAYS USES CHANNEL INPUT SELECTS FOR SAMPLE A
_ADRC=1;           // ADC INTERNAL RC CLOCK
_SAMC=0b11111;     // AUTO SAMPLE TIME = 31TAD (TAD=76ns)
_VCFG=0;           //ADREF+ = AVdd; ADREF- = AVss
_SIMSAM=1;         //SAMPLES CHANNELS SIMULTANEOUSLY
_CHPS = 0b11;      //CONVERTER CANAIS CH0, CH1, CH2, CH3

_CH123NA=0;
_CH123SA=0;        //AN0, AN1, AN2

_SMPI=3;           //INTERRUPÇÃO A CADA 4 CONVERSÕES
_AD1IF=0;          //ZERANDO A FLAG
_AD1IP=PRI_AD;     //DEFININDO PRIORIDADE
_AD1IE=1;          //HABILITANDO A INTERRUPÇÃO
_ADON=1;           //ADC on
}

```

Fonte: Produção do próprio autor.

Código 9 - Arquivo "interrupts.h" para o dsPIC33FJ12MC202.

```

//TIMER 1-----//
//                                     (base de tempo de 0,5segundos)
void _ISR _T1Interrupt(void)
{
IFS0bits.T1IF=0;

    CONTAGEM=TMR2;
    TMR2=0;

    M=M_AD/1023;
    M=M*1.8;
    M=M+0.6;

    Vn=(int) (Vn_AD/10)*10+1000;

    Vmedida=CONTAGEM*M;
    M=M*100;
    Vlimite=Vn*AJUSTEFIXO;

    if(flag_EN)
    {
        if(Vmedida>Vn)
        {
            warning=1;
            if(Vmedida>=Vlimite)    {shutdown=1;}
        }
        else {warning=0;}
    }
}

//TIMER2-----//
//                                     (contator de eventos)
void _ISR _T2Interrupt(void)
{
    _T2IF=0;
}

//INTERRUPÇÃO EXTERNA INT1-----//

```

```

// (botão 1 = ON/OFF PROTEÇÃO)
void _ISR _INT1Interrupt (void)
{
    _INT1IF=0;
    flag_EN=!flag_EN;
}

//INTERRUPÇÃO EXTERNA INT2-----//
// (botão 2 = FAULT RESET)
void _ISR _INT2Interrupt (void)
{
    _INT2IF=0;
    shutdown=0;
}

//CONVERSOR ANALÓGICO DIGITAL DE 10 BITS-----//
void _ISR _ADC1Interrupt(void)
{
    _AD1IF=0;

    SOMAAD1+=ADC1BUF1;
    SOMAAD2+=ADC1BUF2;
    adn++;
    if (adn >= 23)
    {
        adn=0;
        Vn_AD=SOMAAD1/23;
        //Vn=Vn+1000;
        SOMAAD1=0;

        M_AD=SOMAAD2/23;
        //M=M/1023;
        SOMAAD2=0;
    }
}

```

Fonte: Produção do próprio autor.

Código 10 - Arquivo "timer_x.h" para o dsPIC33FJ12MC202.

```

unsigned long int f_base = 1000;
unsigned long int m_base = 1;
unsigned long int tmr3_cont = 0;
//unsigned int tmr3_cont_overflow = 0;

unsigned int dj,dnop;

void config_tmr3 (long int frequencia, long int prescale, int start)
{
    //----->Inicialização<--
    T3CONbits.TON = 0; //desligado
    PR3 = Fcy/(prescale*frequencia);
    TMR3 = 0;
    //T2CONbits.T32 = 0;
    T3CONbits.TGATE = 0;
    T3CONbits.TCS = 0;
    //----->Interrupções<--
    IPC2bits.T3IP = PRI_TMR3;
    IFS0bits.T3IF = 0;
    IEC0bits.T3IE = 1;
    //----->Prescale<--
    if (prescale == 1) //Atribuindo time base input prescale
        T3CONbits.TCKPS = 0b00;
    else if (prescale == 8)
        T3CONbits.TCKPS = 0b01;
    else if (prescale == 64)
        T3CONbits.TCKPS = 0b10;
    else if (prescale == 256)

```

```

        T3CONbits.TCKPS = 0b11;
    else
        T3CONbits.TCKPS = 0b00;
//----->Start<-----
    T3CONbits.TON = start;
}

void tmr3_start (int start)
{
    T3CONbits.TON = start;
}

void delay_ms (int x)
{
    unsigned long int    f_base = 1000;
    unsigned long int    m_base = 1;
    unsigned long int    cont_break;

    cont_break = tmr3_cont + x*m_base;

    while (tmr3_cont != cont_break);
}

void delay_us (int x)
{
    x *= cyc;
    for (dj=0;dj<=x;dj++)
    {
        dnop++;
    }
}

void _ISR _T3Interrupt ()
{
    IFS0bits.T3IF=0; //ZERAR A FLAG
    tmr3_cont++;    //contador geral para "delay_ms()"
}

```

Fonte: Produção do próprio autor.

Código 11 - Arquivo "LCD.h" para o dsPIC33FJ12MC202.

```

//Programa: Biblioteca de escrita em LCD por 4 vias

//Esse código possui as funções básicas para escrita em um display LCD
//funcionando corretamente em um PIC 16F877a. Inclui funções para escrita de
//caractere e string...não realiza operações de leitura, nem está incluso nas
//rotinas a habilitação do modo leitura. Logo, o pino Read/Write (R/W) do LCD
//deve ser sempre aterrado (em hardware).

//Funciona no modo 4 vias, para 'economizar' pinos do microcontrolador.
//Para isso, as rotinas de configuração configuram o display para esperar sempre
//dois nibbles, e então mostrar o char recebido...

//-----//
//          Funções da biblioteca:          //
//-----//
//-->ENVIA_NIBBLE (X);    , onde x é um byte
//-->ENVIA_BYTE (Y,X);    , onde x é o byte enviado, e y especifica o tipo de dado
//comando ou caractere);
//-->ESCREVE_LCD(X);      , onde x é um byte e será interpretado pelo LCD como
//caractere;
//-->COMANDO_LCD(X);      , onde x é um byte e será interpretado pelo LCD como

```

```

comando;
/-->LIMPA_LCD();           , a função não recebe parâmetros, e limpa a tela do LCD;
/-->CONFIG_LCD();          , sem parâmetros, executa as operações padrões de
configuração do LCD;
/-->FRASE(X);              , onde x é um vetor char, e todos os caracteres desse vetor
são enviados ao LCD;
/-->PRINTF(FUNC,"STRING"); , onde func é qualquer função de saída de byte (das
anteriores).
//-----//

//void ESCRIVE_LCD (void);

//#include <stdio.h> //NECESSÁRIO PARA STRLEN();

#define LINE_FEED 0xC0
#define LF 0xC0
#define CR 0x10

//DEFINIÇÕES DE PINOS
#define LCD_RS LATBbits.LATB9 // DEFININDO PINOS DE CONTROLE...
#define LCD_EN LATBbits.LATB8
#define LCD_D4 LATBbits.LATB7 // E DAS 4 VIAS.
#define LCD_D5 LATBbits.LATB6
#define LCD_D6 LATBbits.LATB5
#define LCD_D7 LATBbits.LATB4

#define RS_TRIS TRISBbits.TRISB9 = 0
#define EN_TRIS TRISBbits.TRISB8 = 0
#define D4_TRIS TRISBbits.TRISB7 = 0
#define D5_TRIS TRISBbits.TRISB6 = 0
#define D6_TRIS TRISBbits.TRISB5 = 0
#define D7_TRIS TRISBbits.TRISB4 = 0

//USO DE SAIDAS FIXAS, PARA OTIMIZAR A RAM
//#USE FIXED_IO(D_OUTPUTS = PIN_D0, PIN_D1, PIN_D2, PIN_D3, PIN_D4, PIN_D5, PIN_D6,
PIN_D7)

//#DEFINE PORTF = 0x08

//-----
// ROTINAS DO LCD...
//-----

/-->ENVIA UM NIBBLE (ESTAMOS EM SISTEMA DE 4 VIAS)<--
void ENVIA_NIBBLE (char DADO)
{
    if (DADO & 0x01) LCD_D4 = 1; else LCD_D4 = 0; //CARREGANDO OS PINOS RD4 A RD7
COM UM
    if (DADO & 0x02) LCD_D5 = 1; else LCD_D5 = 0; //NIBBLE. OS 4 BITS MENOS
SIGNIFICATIVOS.
    if (DADO & 0x04) LCD_D6 = 1; else LCD_D6 = 0;
    if (DADO & 0x08) LCD_D7 = 1; else LCD_D7 = 0;

    LCD_EN = 1; // HABILITA ENTRADAS NO LCD
    delay_us(1); //AGUARDA
    LCD_EN = 0; // DESABILITA...
}

/-->ENVIA UM BYTE (DOIS NIBLES). O BYTE PODE SER UM COMANDO OU CARACTERE<--
//SE MODO = 0, ENTÃO O DADO SERÁ INTERPRETADO COMO COMANDO
//SE MODO = 1, ENTÃO O DADO SERÁ INTERPRETADO COMO CARACTERE
void ENVIA_BYTE (int MODO, char DADO)
{
    if (MODO == 1) LCD_RS = 1; else LCD_RS = 0; //CONFIGURA O MODO...
    delay_us(50); //AGUARDA

```

```

    LCD_EN = 0;           //DESABILITA ENTRADA
    ENVIA_NIBBLE (DADO>>4); //ENVIA A PARTE ALTA (ROTATE_RIGHT 4 VEZES)
    ENVIA_NIBBLE (DADO); //ZERA A ALTA E ENVIA A BAIXA...
    delay_us(40);         //AGUARDA
}

//-->SIMPLIFICACANDO O ENVIO...NESSE CASO, SÓ SE ENVIA CARACTERE<--
void ESCRIVE_LCD (char C)
{
    ENVIA_BYTE(1,C);
}

//-->E NESSE CASO, SÓ ENVIA COMANDOS<--
void COMANDO_LCD (char C)
{
    ENVIA_BYTE(0,C);
}

//-->COMANDO PARA LIMPAR A TELA<--
void LIMPA_LCD()
{
    ENVIA_BYTE (0,0x01); // COMANDO CLEAR DO DISPLAY
    delay_ms (2);
}

void ESCRIVE_NUM (int num)
{
    ESCRIVE_LCD(num + 48);
}

void LCD_CR () //cariage return
{
    COMANDO_LCD (CR);
}

void LCD_BS () //backspace
{
    COMANDO_LCD (CR);
    ESCRIVE_LCD (' ');
    COMANDO_LCD (CR);
}

//-->ROTINA DE CONFIGURAÇÃO<--
void CONFIG_LCD()
{
    RS_TRIS; //DEFININDO SAIDAS DE ESCRITA
    EN_TRIS;
    D4_TRIS;
    D5_TRIS;
    D6_TRIS;
    D7_TRIS;

    LCD_D4 = 0;
    LCD_D5 = 0;
    LCD_D6 = 0;
    LCD_D7 = 0;
    LCD_RS = 0; //PREPARA ESCRITA DE COMANDO DE CONFIGURAÇÃO
    LCD_EN = 0;
    delay_ms(15);
    ENVIA_NIBBLE(0x03); //INICIALIZA DISPLAY
    delay_ms (5);
    ENVIA_NIBBLE (0x02); //COMANDO 'CURSOR HOME' DO DISPLAY
    delay_ms (1);
    ENVIA_BYTE(0,0x28); //CONFIGURA O LCD PARA OPERAR EM 4 BITS, EM DUAS LINHAS.
    ENVIA_BYTE(0,0x0c); //DISPLAY LIGADO E SEM CURSOR
    LIMPA_LCD(); //CLEAR
    ENVIA_BYTE(0,0x06); //OS DADOS DESLOCAM PARA A DIREITA
}

```

```

//-----

void FRASE (char *STRING)
{
    int LEN;
    int I;
    LEN = strlen(STRING);
    I = LEN;
    while (LEN)
    {
        ESCRIVE_LCD(STRING[I - LEN]);
        LEN--;
    }
}

//-----TESTANTOOO
void EN_LCD (int n, char ndig, char mode, char punkt)
{
    //n = numero a ser escrito
    //ndig = numero de digitos
    //mode = operação com sinal (+ e -) ou sem sinal

    unsigned int i,r,num;

    if(mode==1){
        if(n<0){
            ESCRIVE_LCD('-');
            num = (unsigned int)(-n);
        }
        else{
            ESCRIVE_LCD('+');
            num = (unsigned int)n;
        }
    }
    else{
        num=(unsigned int)n;
    }

    r = num;
    i = (unsigned int)(r / (unsigned int)(10000));
    if(ndig==5) ESCRIVE_LCD(i+48);
    if(punkt==4) ESCRIVE_LCD('.');
    r = r - (unsigned int)(i*(unsigned int)10000);
    i = r / (1000L);
    if(ndig>3) ESCRIVE_LCD(i+48);
    if(punkt==3) ESCRIVE_LCD('.');
    r = r - (unsigned int)(i*(unsigned int)1000);
    i = r / (unsigned int)100L;
    if(ndig>2) ESCRIVE_LCD(i+48);
    if(punkt==2) ESCRIVE_LCD('.');
    r = r - (unsigned int)(i*(unsigned int)100);
    i = r / 10;
    if(ndig>1) ESCRIVE_LCD(i+48);
    if(punkt==1) ESCRIVE_LCD('.');
    r = r - (i*(unsigned int)10);
    ESCRIVE_LCD(r+48);
}

void linha_lcd(int linha)
{
    if (linha == 1)
        COMANDO_LCD(0x80);
    else if (linha == 2)
        COMANDO_LCD(0xC0);
}

```

```

}

void posicao_lcd (int linha, int coluna)
{
    if (linha == 1)
        COMANDO_LCD(0x7F+coluna);
    if (linha == 2)
        COMANDO_LCD(0xBF+coluna);
}

```

Fonte: Produção do próprio autor.

Código 12 - Arquivo "main.c" para o dsPIC33FJ12MC202.

```

#include <p33FJ12MC202.h>

#include <stdio.h>
#include "main.h"
#include "config_interrupts.h"
#include "interrupts.h"
#include "timer_x.h"
#include "LCD.h"

int main()
{
    init ();
    config_interrupts();
    config_tmr3(f_base,1,1);

    CONFIG_LCD();
    delay_ms(10);

    RELE=0;

    while (1)
    {

        displayn++;

        if(shutdown)
        {
            RELE=1;
            LIMPA_LCD();
            delay_ms(500);
            posicao_lcd(1,1);
            FRASE ("!!!!OVERSPEED!!!!");
            delay_ms(500);
        }
        else
        {
            RELE=0;
            if (displayn>100000)
            {
                displayn=0;

                posicao_lcd(1,1);
                FRASE ("Vn:");
                posicao_lcd(1,4);
                EN_LCD(Vn,4,0,0);
                posicao_lcd(1,8);
                FRASE ("rpm");

                posicao_lcd(2,1);
                FRASE ("V:");
                posicao_lcd(2,3);
                EN_LCD(Vmedida,4,0,0);
                posicao_lcd(2,7);
                FRASE ("rpm");
            }
        }
    }
}

```

```
        posicao_lcd(1,11);  
        FRASE ("M:");  
        posicao_lcd(1,13);  
        EN_LCD (M, 3, 0, 2);  
  
        posicao_lcd(2,10);  
        FRASE ("PROT:");  
        if(flag_EN) {FRASE ("ON");}  
        else {FRASE ("OFF");}  
    }  
}  
  
} //FIM DO WHILE  
} //FIM DA FUNÇÃO MAIN
```

Fonte: Produção do próprio autor.